AD NUMBER
AD814971
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; MAY 1967. Other requests shall be referred to Air Force Weapons Laboratory, Attn: WLRT, Kirtland AFB, NM 87117. This document contains export-controlled technical data.**
AUTHORITY
AFWL ltr, 30 Nov 1971

The following notice applies to any unclassified (including originally classified and now declassified) technical reports released to "qualified U.S. contractors" under the provisions of DoD Directive 5230.25, Withholding of Unclassified Technical Data From Public Disclosure.

NOTICE TO ACCOMPANY THE DISSEMINATION OF EXPORT-CONTROLLED TECHNICAL DATA

1. Export of information contained herein, which includes, in some circumstances, release to foreign nationals within the United States, without first obtaining approval or license from the Department of State for items controlled by the International Traffic in Arms Regulations (ITAR), or the Department of Commerce for items controlled by the Export Administration Regulations (EAR), may constitute a violation of law.

2. Under 22 U.S.C. 2778 the penalty for unlawful export of items or information controlled under the ITAR is up to ten years imprisonment, or a fine of $1,000,000, or both. Under 50 U.S.C., Appendix 2410, the penalty for unlawful export of items or information controlled under the EAR is a fine of up to $1,000,000, or five times the value of the exports, whichever is greater; or for an individual, imprisonment of up to 10 years, or a fine of up to $250,000, or both.

3. In accordance with your certification that establishes you as a "qualified U.S. Contractor", unauthorized dissemination of this information is prohibited and may result in disqualification as a qualified U.S. contractor, and may be considered in determining your eligibility for future contracts with the Department of Defense.

4. The U.S. Government assumes no liability for direct patent infringement, or contributory patent infringement or misuse of technical data.

5. The U.S. Government does not warrant the adequacy, accuracy, currency, or completeness of the technical data.

6. The U.S. Government assumes no liability for loss, damage, or injury resulting from manufacture or use for any purpose of any product, article, system, or material involving reliance upon any or all technical data furnished in response to the request for technical data.

7. If the technical data furnished by the Government will be used for commercial manufacturing or other profit potential, a license for such use may be necessary. Any payments made in support of the request for data do not include or involve any license rights.

8. A copy of this notice shall be provided with any partial or complete reproduction of these data that are provided to qualified U.S. contractors.

DESTRUCTION NOTICE

For classified documents, follow the procedure in DoD 5220.22-M, National Industrial Security Program, Operating Manual, Chapter 5, Section 7, or DoD 5200.1-R, Information Security Program Regulation, Chapter 6, Section 7. For unclassified, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.

NUCLEAR EXPLOSION INTERACTION STUDIES

Volume III

Miscellaneous Code Development

K. D. Pyatt, Jr., et al.

General Atomic Division
General Dynamics Corporation
Special Nuclear Effects Laboratory
San Diego, California

Contract AF 29(601)-7035

TECHNICAL REPORT NO. AFWL-TR-66-108, Vol. III
May 1967

AIR FORCE WEAPONS LABORATORY
Research and Technology Division
Air Force Systems Command
Kirtland Air Force Base
New Mexico

AFWL-TR-66-108, Vol. III

Research and Technology Division
AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base
New Mexico

NUCLEAR EXPLOSION INTERACTION STUDIES

Volume III

Miscellaneous Code Development

K. D. Pyatt, Jr., et al.

General Atomic Division
General Dynamics Corporation
Special Nuclear Effects Laboratory
San Diego, California
Contract AF29(601)-7035

TECHNICAL REPORT NO. AFWL-TR-66-108, Vol. III

FOREWORD

This report was prepared by General Atomic Division, General Dynamics Corporation, San Diego, California, under Contract AF29(601)-7035. The research was funded by DASA under Program Element 6.16.46.01D, Project 5710, Subtask 07.002, and by ARPA Order 313, Program Element 6.25.03.01.R.

Inclusive dates of research were 22 July 1965 to 21 July 1966. The report was submitted 28 April 1967 by the AFWL Project Officer, Maj George Spillman (WLRT). The contractor's report number is GA-7370.

This final report on Nuclear Explosion Interaction Studies is being published in four volumes. The volume titles are as follows: Volume I, Methods for Analysis of Radiative Transfer; Volume II, Methods for Analysis of Thermal Phenomena; Volume III, Miscellaneous Code Development; and Volume IV, Phenomenology Studies (classified SECRET/RESTRICTED DATA).

The first three volumes are devoted, respectively, to theoretical studies and computer code development in radiative transfer, thermal phenomena, and miscellaneous efforts related to various other aspects of the work. The fourth volume, which is classified, contains the results of applications of these techniques, and of those previously developed, to the study of fireball growth and the interaction of laser radiation with materials.

The NEIS program is long-range, and most of the projects described in this report are in an incomplete state of development. This is due in part to the nature of the existing computer programs themselves, which continue in a state of development as long as they are in use, and in part to the time scale involved in bringing new programs to a state of capability for solving real problems.

General Atomic staff personnel contributing to the research include J.H. Alexander, C.R. Dismukes, R. Durstenfeld, R.S. Engelmore, B.E. Freeman, W.B. Lindley, J.T. Palmer, K.D. Pyatt, L.L. Reed, L.M. Schalit, J.R. Triplett, and numerous others. The cooperation of Col R.H. Pennington, Maj G.R. Spillman, Lt B.S. Chambers, III, Lt N.D. Morgan, Lt R.A. Howerton, Dr. P.V. Avizonis, and Mr. D.W. Lane of the Air Force Weapons Laboratory is gratefully acknowledged.

This report has been reviewed and is approved.

George R. Spillman

GEORGE R. SPILLMAN
Major, USAF
Project Officer

George R. Spillman

for RALPH H. PENNINGTON
Colonel, USAF
Chief, Theoretical Branch

CLAUDE K. STAMBAUGH
Colonel, USAF
Chief, Research Division

ABSTRACT

A group of SPUTTER routines has been modified or developed to execute in a more nearly accurate and efficient manner the task of "rezoning" the Lagrangian zone structure in a specified region of the system being analyzed. The automatic rezoning routine AUTORZ has been expanded to provide for maintaining a prescribed spatial definition within a region. The control routines REZONE, CONVER, and ZONE have been modified to make use of the new routines ZMS and ZRAD, which accomplish conversion between space and mass coordinates, and a special procedure PROFL is described which performs energy interpolation in certain cases. To facilitate maintenance of complex computer codes such as SPUTTER, OIL, and HECTIC, two FORTRAN analysis routines (FAR, Programs One and Two) have been developed and are nearly complete at this time. These routines are designed for replacement of FORTRAN symbol strings by others, and for compilation of complete cross reference listings of variable and subroutine references. Editing routines designed for specific use with the data-link terminal facilities now installed at the General Atomic Special Nuclear Effects Laboratory (SNEL) and at AFWL are also described.

This page intentionally left blank.

CONTENTS

DOCUMENTS PRODUCED UNDER CONTRACT AF 29(601)-7035

Alexander, J. H., "Numerical Methods for Solving Two-Dimensional Problems of Radiative Transport," General Atomic Report GAMD-7047, 1966. (Unclassified)

Freeman, B. E., "Polarization Effects on Radiative Diffusion in a Compton Scattering Medium," General Atomic Report GA-6157, 1966, submitted for publication in Astrophys. J.

Palmer, J. T., "A Comparison of Various Difference Schemes for a Simplified One-Dimensional Transport Equation," General Atomic Report GAMD-6955, 1966. (Unclassified)

"Progress Report for Period Ending September 30, 1965" (U), General Atomic Report GACD-6747(9-65), 1965. (SECRET)

"Progress Report for Period Ending December 31, 1965," General Atomic Report GACD-6747(12-65), 1965. (Unclassified)

"Progress Report for Period Ending March 31, 1966," General Atomic Report GACD-6747(3-66), 1966. (Unclassified)

"Progress Report for Period Ending June 30, 1966," General Atomic Report GACD-6747(6-66), 1966. (Unclassified)

Reed, L. L., "Equation of State Inversion from Temperature-Density to Specific Energy - Density by Table Look-Up," General Atomic Report GAMD-7189 (Rev.), 1966. (Unclassified)

Schalit, L. M., and D. R. Yates, "LPLOT Fortran Program," General Atomic Report GAMD-7104, 1966. (Unclassified)

Teichmann, T., "Numerical Integration on a Sphere," General Atomic Report GAMD-6981, 1966. (Unclassified)

Triplett, J. R., "Heating of a Slab by a Time-Dependent Source," General Atomic Report GAMD-6665, 1965. (Unclassified)

SECTION I

GENERATOR-REZONE ROUTINES FOR SPUTTER

In response to the needs of several applications, extensive modifications have been made in most of the SPUTTER routines to establish or revise the zone structure which partitions the material being studied into convenient slices for computational purposes. The more fundamental of these changes are discussed below.

1.1. AUTORZ (Automatic Rezone Specification)

1.1.1. Introduction

The purpose of this routine is to maintain, during the course of a calculation, an adequately fine-grained zone structure in those portions of the material being studied which are of principal physical interest, and at the same time avoid wasteful use of small zones in other portions. The routine is presently adapted to a restricted class of problems, namely, those in which an ablation front is moving into solid material on the left, with vapor expanding to the right. Only a few large zones are needed to describe the conduction front in the solid material, whereas the zone at the ablation front and the zones in the vapor for a certain distance outward must be much smaller if the objectives of the calculation are to be achieved. The outermost portions of the vapor are of less interest and can be described by a less detailed zone structure.

Since SPUTTER is a Lagrangian code, the zone boundaries move with the material and the problem is particularly serious. Even in an Eulerian code, however, a similar problem arises unless a moving coordinate system can be used with respect to which the ablation front is stationary. The

techniques described in this section may easily be generalized to accommodate problems associated with a wide variety of situations other than those associated with ablation; near shock and radiation fronts, for example, zone definition may be required which is difficult or impossible to maintain without repeated redefinition of the zone structure.

AUTORZ is called at the beginning of each cycle if the flag EKLM (19) is nonzero. The AUTORZ option control, THETAK (58), must also be nonzero or there is an immediate return.

1.1.2. The Solid Rezone Procedure

In the solid material it is necessary to set up a few zones with a very large mass available for ablation and for containing the conduction front, and to split these, one at a time, into small zones as the ablation front advances into the solid. It is assumed that the large zones initially generated were equal in mass, although this is not strictly essential. The splitting ratio is an input quantity, THETAK(53), and is typically set at a number like 10.0 or 20.0. It must be an integer but has REAL type. The index of the boundary between the large zones and the small ones formed by the splitting operation is stored as THETAK(54); this is reduced by unity each time a large zone is split. It is usually best to generate the problem with a "lambda region" of small zones already present, in which case THETAK(54) should be initially set to the left boundary index of this lambda region.

The solid rezone is executed when IA, the ablation front index, has been reduced to the value THETAK(54)+THETAK(57). THETAK(57) is an input specification, usually of the order of 3 to 10, which establishes a minimum for the number of small zones behind the ablation front. This is advisable for reasons involving the conduction and source deposition calculations in some applications.

1.1.3. The Inner Vapor Rezone Procedure

In the analysis of certain experimental arrangements, such as plasma probe measurements, good zone definition must be maintained within a

region of space defined in the Eulerian sense, i.e., in terms of fixed coordinates rather than moving Lagrangian zone boundaries. The region of interest is assumed to extend from the ablation front on the left to a coordinate position specified by the input in THETAK(59). Within this region, the desired zone width is specified by THETAK(60). If THETAK (59)=0 or if the leading edge of the vapor has not yet passed beyond THETAK(59), no attempt to rezone the inner vapor is made. The index K of the zone which contains the outer boundary THETAK(59) of the region of interest and the index J of the outermost zone within this region whose width is less than THETAK(60) are then identified. Unless the solid rezone is being done at the same time, the inner vapor rezone is omitted when the width of zone K is less than 1.5 times THETAK(60). If the solid rezone is to be done anyway, the inner vapor rezone is also done provided that there is at least one zone between J and K.

The inner vapor rezone is specified as follows: left boundary index IL=J+1, right boundary index IR=K+1, left zone width -VAL(2)=THETAK(60), number of zones VAL(5)=one more than the minimum number for which none of the new zones would exceed THETAK(60) in width. This specification generates new zones whose width decreases slightly toward the right, a desirable situation since it compensates in advance for later expansion of the zones.

1.1.4. The Outer Vapor Rezone

In order to prevent the total number of zones from increasing too rapidly due to repeated solid rezoning, the outer part of the vapor is rezoned periodically to increase zone size in this region. The outer region extends from the outer boundary of the "inner vapor region" to the leading edge of the vapor, R(IB). The inner vapor region consists of the rezone region defined in the preceding paragraph, if such a rezone is being done; otherwise it consists of a definite (input) number THETAK(55) of zones counted out from the ablation front. The outer vapor rezone is normally done whenever the solid or inner vapor rezones are done; however, if the inner vapor is

being rezoned, the outer region must have a minimum of THETAK(55) zones in it or it is not considered worthwhile.

The outer region is rezoned as follows: left-most zone mass VAL(2) =the mass of the outermost of the inner zones, ratio of each zone mass to its neighbor on the left VAL(4)=THETAK(56). (In order to reduce the number of zones, THETAK(56) must be greater than 1.)

1.1.5. Additional Controls and Flags

If THETAK(58)=1., only solid rezones can be done; if THETAK(58)= -1., solid rezones are possible only on cycles for which both inner and outer vapor rezones are also done. For THETAK(58)=2., there are no controls beyond those listed in the preceding paragraphs. If a solid rezone is to be done, the temperature profile flag is set (this is a minus sign on the right boundary index IR for the solid rezone specification). If any rezones are to be done, the linkage flag S16 is set to -1, indicating to the rezone programs that they are to rezone from data in RK, and MAIN1 is then called.

1.1.6. The RK Array

The RK array contains specifications for a maximum of nine rezone regions, which the rezone program will process in the order of placement in the RK array. Thus, AUTORZ specifies that the solid rezone is to be done after the vapor rezones by listing its specifications after those for the vapor rezones (if any).

RK(1) contains the number of rezone regions specified in the array. Each specification consists of 11 words, i.e., RK(2)-RK(12), RK(13)-RK(23), etc. These words are, respectively:

IL	Left boundary index of region (REAL type)
IR	Right boundary index of region (REAL type)
VAL(2)	Mass (if positive) or width (if negative) of left-most zone
VAL(3)	Mass (if positive) or width (if negative) of right-most zone

VAL(4)	Width ratio (if VAL(6)=1) or mass ratio (if VAL(6)=2) of each zone to its neighbor on the left
VAL(5)	Number of zones to be generated in the region
VAL(6)	Mass-space switch (see VAL(4))
VAL(7)	Material identification number
VAL(8)	Multiplicative factor for opacity
VAL(9)	$\overline{A}$ (mean atomic mass number)
VAL(10)	ZMEAN (mean atomic number)

Of these 11 parameters, those required are the first two, at least two of the next four, and the seventh (VAL(6)). The others can either be calculated from those given or retain their previous values unchanged. The AUTORZ routine always leaves the last four words unspecified.

1.2. REZONE (REZONE CONTROL ROUTINE)

On entry to this routine, rezone specifications have been placed in the RK array either by AUTORZ or by the ZONERD routine which reads the specifications from card input. The number of such specifications is given by RK(1), and the entire procedure described below is repeated for each specification in turn. The format for the specifications is given in Section 1.1.6.

The rezone procedure will modify information within the rezone region defined by the boundary index range $IL \leq I \leq IR$. Unmodified values must first be saved, as follows:

	Quantity		Save location	
For all I, IL≤I≤IR	Velocities, RD(I)		RDTAB(N)	in /NAMEC/ COMMON
	Densities, 1./SV(I)		RHOTAB(N)	
	Specific energies, E(I)		THETAB(N)	
	Coordinates, R(I)		RTABRD(N)	
	Coordinates, R(I)		RTABRO(N)	
	Coordinates, R(I)		RTABTH(N)	
	Mean ionic charges, FEW(I)		CAR(N)	in blank COMMON
	Z^1, X6(I)	Elemental mean ionic charges	CAPAC(N)	
	Z^2, BC(I)		CHIC(N)	
	Z^3, BR(I)		CHIR(N)	
	Z^4, CRTC(I)		X4(N)	
	Z^5, RHO(I)		X5(N)	
	Zone masses, G(I)		EC(N)	
	Mass depletion energies, RFF(I)		PB(N)	
	Mass depletion area, SMLR(I)**2		SMLQ(N)	
	Left boundary of rezone region, R(IL)		RR	/NAMEC/ or internal
	Right boundary of rezone region, R(IR)		SAVRV6	
	Pusher velocity RD(1) or RD(IG-1)		RLF	
	Solid-vapor interface (see below)		RPUSH	
	LTE-non-LTE interface R(NR)		RNR	
	Left boundary of active region R(ICA)		RICA	
	Right boundary of active region R(ICB)		RICB	

The solid-vapor interface is defined by R(IA) for a left pusher (solid region) or R(IB) for a right pusher, provided that the vaporization flag BOILA is zero. If BOILA $\neq$ 0, RPUSH is located within the vaporizing zone at a position defined by the fraction of the mass in the zone which is still unvaporized. When the new zone structure is established, these mass fractions are recomputed if necessary.

The subroutine GENRAT is then called to fill in any previously unspecified quantities VAL(2), (3), (4), (5), and subroutine MOVE is called to shift blank COMMON array elements with indices I = IR to their new locations. The rezone index IR for the current rezone region and the indices IL and IR in the RK array for yet unprocessed regions to the right of the current region are also shifted. Other indices and quantities which have been saved as described above are shifted to their new values. The ZONE routine (see below) generates new boundary positions, masses, and specific volumes in the rezone region. Boundary velocities are recomputed by linear spatial interpolation.

The next step is to recompute, for the new zones, the quantities E(I), FEW(I), RDD(I) SMLR(I), and the five elemental mean ionic charges. This is done by mass-weighting (except for RDD) to produce corresponding extensive quantities, computing the amount of each of these quantities in each subzone (subzones are defined by the union of the old and the new zone boundaries), assembling the subzone quantities belonging to each of the new zones, and dividing by the new zone masses (except for RDD) to restore the original physical dimensions. Subroutine UP37A is called to update lambda-region arrays and clear the "save" locations of the old information which is no longer needed.

Next, subroutine "PROFL" (see below) is called if its flag was set, to eliminate by a special procedure any plateaus in the specific energy distribution in the solid which result from zone splitting. Finally, for LTE zones, the temperature and all of the thermodynamic variables except the specific internal energy are evaluated by an inversion of the state equation.

If the material identification number is in the range 201 to 300, this requires an iterative procedure, using interval halving or similar techniques. For non-LTE zones and LTE zones which specify the general ionic state equation routine EIONX (by material identification numbers ≤ 200 or > 300), a more efficient procedure is being developed which makes use of the established elemental mean ionic charges and requires no iteration. At present, a simple version of this new procedure is used for non-LTE zones in one-material problems.

1.3. CONVER, ZONE, ZRAD, AND ZMS

The most important mathematical problem encountered in generating or rezoning a problem is the conversion between spatial coordinates and mass coordinates. The relationship between these is specified by a density profile, which is just a list of material densities for each of the original zones (in the rezoning case), or is given by input in generating a new problem. The four routines described in this section are the ones involving this conversion process, and are therefore discussed together.

Zoning may be carried out either with space or mass as the basic independent parameter. If VAL(6), the mass-space switch, is 1., the zones are to be set up with a uniform ratio VAL(4) of spatial widths, while VAL(6)=2. specifies a uniform ratio VAL(4) of zone masses. With space zoning, the conversion problem is one of integrating the density profile to obtain new zone masses, whereas with mass zoning, the problem is to integrate the reciprocal of the density profile to obtain new zone widths. The problem is nontrivial for three reasons:

1. The space variable may be the radius variable of a cylindrical or spherical coordinate system, in which case the interpretation to be given the density profile is not without ambiguity.
2. The zone widths must, in many cases, add up exactly to a value (the total region width) prescribed in advance, and the new boundary positions must conform to the zone widths as well as

possible within the limits of the finite computer word size. The sequences of boundary positions and zone widths need not be strictly derivable from each other, but they must both be regular and monotonic, with initial and terminal terms fixed. In some applications the boundary coordinates and zone widths are 8 to 10 orders of magnitude apart, and numerical methods designed for easier problems do not work.

3. For reasons of efficiency, neither involved iterative procedures nor indiscriminate use of double-precision operations is an acceptable approach to these problems.

The interpretation given the density profile in generating a new problem is that the complete density function is to be obtained by interpolating between the given points, and that the interpolant is linear not in the coordinate r, but in r^{α} where α=1,2,3 for plane, cylindrical, and spherical symmetry respectively. This eliminates the need for solving anything more complicated than a quadratic equation in finding the width of a zone given the mass. In rezoning, the density profile is in origin a discrete mass profile, and any interpolation at all would be in a sense gratuitous. The assumption which is made throughout SPUTTER is that the density is constant throughout each zone; in rezoning, therefore, the profile is considered as defining a histogram, and the integration process is thus always linear. The integration subroutines ZMS and ZRAD are directed to these two alternative interpretations of the density profile by a switch IZSWT which is set in MAIN1 to -1 for rezoning and to 0 for generating.

The union of the set of new zone boundaries and the set of old zone boundaries or profile points thus defines a set of subzones within each of which the density is either constant or linear in r^{α}. The mass integration routine ZMS is called for each subzone to determine the mass contained therein. The inverse problem is solved by the ZRAD subroutine, in which

each subzone has a given mass and it is required to find the volume from the profile. In these routines the evaluation of the slope from the profile is done in single precision, while the integration calculation itself may be done in double precision.

The CONVER routine is called from GENRAT to convert specifications into consistent form. Thus, if VAL(6)=2. specifying mass zoning, but any of VAL(1), VAL(2), or VAL(3) are specified with negative sign to indicate that they are widths, not masses, then the CONVER routine, with the aid of ZMS and the density profile, computes the proper masses where necessary for VAL(1) (region mass), VAL(2) (first zone mass), and VAL(3) (last zone mass). Similarly, CONVER calls ZRAD in the less usual case where space zoning is specified in terms of the mass of the region or of the first or last zones.

The ZONE routine computes the mass, width, boundary coordinates, and specific volume of each zone. After generating the zones using VAL(4) as the generating ratio, the subzone partition is established and ZMS (for space zoning) or ZRAD (for mass zoning) is called for each in turn to complete the description of the subzone. The assemblage of the subzones into the new zones to which they belong may be done with double numerical precision. The largest discrepancies arise for the boundary positions, which are cumulative. If the right region boundary is prescribed (always the case in rezoning and often the case in generating), the generated region will exceed or fall short of the target by an amount which is not necessarily small compared with the width of the last zone, even if double precision operations as described above are used. Smaller discrepancies arise for the width or mass of the last zone, since the prescribed value VAL(3) will not, in general, agree to the last digit with the value resulting from starting with VAL(2) and multiplying N times by VAL(4).

In each of these cases the discrepancies are removed by the following Procrustean device: The quantity in question is regenerated in reverse order (from right to left), using the prescribed right-end value as the initial condition. In calculating widths or masses, the initial condition is thus

VAL(3) and the generating ratio is 1./VAL(4), while in calculating boundary positions, the initial value is R(IR) and the increment is -DELTAR(I), where DELTAR(I) is the same as the value used in the forward sweep. Each quantity obtained in the forward sweep is then multiplied by a factor W(I) and each reverse-sweep quantity by 1.-W(I), and the two results are then added to give the final result. W(I) is defined as that linear function of the index I which assumes the value 1. at the left end of the region and 0. at the right end.

It may be the case that this last procedure eliminates the need for double precision arithmetic at any stage of the calculation. Experimentation on this point is planned.

1.4. PROFL

The PROFL subroutine is designed to eliminate a severe perturbation of the solid heat conduction calculation which results from splitting a single large zone into 10 or 20 small zones, each with the same specific internal energy and therefore the same temperature. The effect of this is to multiply the conductive flux at the ends of this region by a large factor and to destroy it entirely in the interior.

It is assumed that the conduction front is moving to the left, so that the temperature is a nondecreasing function of position r. The profile to be established is required to assume the same values at the ends of the region as the adjacent zones outside, and to conserve the original total energy content. These three conditions are sufficient to determine a quadratic function but not to ensure that the quadratic is a non-decreasing function of r. The profile is therefore defined as a quadratic which, starting at the right boundary, decreases monotonically, reaching the left boundary value at a point which may be in the interior. The profile in that case is continued as a constant to the left boundary. The energy in each zone is then simply the integral of this profile across the zone.

The geometry may have plane, cylindrical, or spherical symmetry.

1.5. APPENDIX: CODE LISTINGS

```
      SUBROUTINE AUTORZ                                                  AUT00010
C                                                                        AUT00020
C     COMPILED MAY 16, 1966                                              AUT00030
C                                                                        AUT00040
C                                                                        AUT00050
C*************************************************************** ******AUT00060
C                                                                        AUT00070
C*                    S P U T T E R   C O M M O D                       **AUT00080
C*                                                                      **AUT00090
C*                                                                      **AUT00100
      COMMON   LMDA(37), NR    , NSMLR , IA    , ID    , ICA   , ICD   ,AUT00110
     1   KMAX  , BLANK1, BLANK2, BLANK3, IAP1  , IDP1  , ICAP1 , ICDP1 ,AUT00120
     2   II    , IG    , NRAD  , BLANK4, IAM1  , IDM1  , ICAM1 , ICDM1 ,AUT00130
     3   IIP1  , IGM1  , IALPHA, BLANK5, TH    , TMAX  , BLANK6, DELPRT,AUT00140
     4   FREQ  , CNTMAX, AR    , ASMLR , PUSHA , PUSHB , BOILA , BOILB ,AUT00150
     5   CVA   , CVB   , SLUG  , ALPHA , HVA   , HVB   , HCA   , HCB   ,AUT00160
     6   EMINA , EMINB , CA    , CB    , GA    , GB    , GL    , GR    ,AUT00170
     7   RHOL  , RHOR  , EPIO  , EPSI  , RIA   , RIB   , RDIA  , RDIB  ,AUT00180
     8   RPIA  , RPIB  , RPDIA , RPDIB , TPRINT, TA    , TB    , TC     AUT00190
      COMMON     TD    , TE    , DTH2  , DTH2P , DTH1  , DTRMIN, DTMAX ,AUT00200
     1   DTMAX1, DTMAX2, DTMAX3, DTR   , SWITCH, CO    , CMIN  , DELTA ,AUT00210
     2   GAMA  , WCRIT , SIGMAQ, AC    , ACOSTA, CNVRT , SUMPA , SUMRS ,AUT00220
     3   ROIA  , ROIAM1, ROIB  , ROIBP1, GMS   , S1    , S2    , S3    ,AUT00230
     4   S4    , S5    , S6    , S7    , S8    , S9    , S10   , S11   ,AUT00240
     5   S12   , S13   , S14   , S15   , S16   , S17   , S18   , S19   ,AUT00250
     6   S20   , EO    , FO    , TAU   , ZERO  , R   (152), DELTAR(152),AUT00260
     7  ASQ   (152), RD    (152), VD    (152), RDD   (152), SMLR  (152),AUT00270
     8  DELR  ( 37), P     (152), P1    (152), PB    (152), PB1   (152) AUT00280
      COMMON          P2    (152), SV    (152), RDO   (152), THETA (152),AUT00290
     1  W     (152), E     (152), EI    (152), EK    (152), A     (152 ,AUT00300
     2  V     (152), G     (152), D     (152), C     (152), X2    (152),AUT00310
     3  X3    (152), X4    (152), X5    (152), X6    (152), X7    (152),AUT00320
     4  SMLA  (152), SMLB  (152), SMLC  (152), SMLD  (152), SMLE  (152),AUT00330
     5  EC    (152), ER    (152), SMLG  (152), SMLH  (152), BIGA  (152),AUT00340
     6  BIGB  (152), CV    (152), BC    (152), BR    (152), CHIC  (152),AUT00350
     7  CHIR  (152), CAPAC (152), CAPAR (152), CRTC  (152), CRTR  (152),AUT00360
     8  CRTPC (152), GOFR  (152), FEW   (152), CAR   (152), OKLM  ( 37) AUT00370
      COMMON          TELM  ( 37), EKLM  ( 37), ELM   ( 37), FCLM  ( 37),AUT00380
     1  FRLM  ( 37), WLM   ( 37), QLM   ( 37), AMASNO( 37), CHRNO ( 37),AUT00390
     2  ZP1   ( 37), ZP2   ( 37), SOLID ( 37), ECHCK ( 37), RK    (104),AUT00400
     3  RL    ( 37), RHOK  (104), RDK   (104), THETAK(104), TEMP  ( 16),AUT00410
     4  HEAD  ( 12), MAXL        , MAXLM                                 AUT00420
C                                                                        AUT00430
C*                                                                      **AUT00440
C*************************************************************************AUT00450
C                                                                        AUT00460
C      ************ INPUT DATA ***************                          AUT00470
C     THETAK(53)=SOLID REZONE RATIO                                      AUT00480
C     THETAK(54)=SOLID REZONE INTERFACE                                  AUT00490
C     THETAK(55)=NUMBER OF SMALL VAPOR ZONES                             AUT00500
C     THETAK(56)=SSIGMA IN VAPOR TO RIGHT OF INTERFACE                   AUT00510
C     THETAK(57)=REZONE CRITERION                                        AUT00520
C     THETAK(58) = 1. FOR SOLID REZONE ONLY, = 2. FOR SOLID AND VAPOR    AUT00530
C     THETAK(58) .EQ.-1. FOR REZONE IN VAPOR ONLY
C     THETAK(59) = RT. BOUNDARY FOR CONSTANT ZONES
C     THETAK(60) = CONSTANT ZONE SIZE
      IF (THETAK(58).EQ.0.) GO TO 1000
      ISR=THETAK(54)
      NREZ=THETAK(57)                                                    AUT00540
      NSVZ = THETAK(55)                                                  AUT00550
      IF (IA.LE.ISR+NREZ) GO TO 10
      IF (THETAK(58).EQ.1.) GO TO 1000
      IF (THETAK(59).EQ.0.) GO TO 1000
      IF (R(IB).LT.THETAK(59)) GO TO 1000
      RZCF = 1.
      GO TO 11
   10 RZCF = 0.
      THETAK(54)=ISR-1
                                                                         AUT00580
```

```
      IF (THETAK(58).EQ.1.) GO TO 100
      IF (THETAK(59).EQ.0.) GO TO 20
   11 K = 0
      DO 12 I = IA, IB
      J = IA + IB - I
      IF (R(J).GT.THETAK(59)) GO TO 12
      IF (K.EQ.0) K = J
      IF (DELTAR(J).LT.THETAK(60)) GO TO 13
   12 CONTINUE
C     K = RT. INDEX FOR CONSTANT ZONES
C     J = LF. INDEX FOR CONSTANT ZONES
C      - -  FULL REZONE  - -
   13 IF (RZCF.GT.0. .AND. DELTAR(K).LT.1.5*THETAK(60)) GO TO 1000
      IF (RZCF.EQ.0. .AND. K-J.LT.3) GO TO 20
      RK(2) = J + 1
      RK(3) = K + 1
      RK(4) = -THETAK(60)
      RK(5) = 0.
      RK(6) = 0.
      RK(7) = INT((R(K+1)-R(J)) / THETAK(60)) + 1
      RK(8) = 1.
      IF (IB.GE.(K + 1 + NSVZ)) GO TO 14
      IF (RZCF.EQ.1.) GO TO 499
      GO TO 21
   14 RK(13) = K + 1
      RK(14) = IB
      RK(15) = G(K)
      RK(16) = 0.
      RK(17) = THETAK(56)
      RK(18) = 0.
      RK(19) = 2.0
      IF (RZCF.EQ.1.) GO TO 40
      RK(24) = THETAK(54)
      RK(25) = -ISR
      RK(26) = G(ISR-1) / THETAK(53)
      RK(27) = RK(26)
      RK(28) = 1.0
      RK(29) = THETAK(53)
      RK(30) = 2.0
      RK(1) = 3.
      GO TO 500
C      - -  OUTER VAPOR FIRST  - -
   20 IL = IA + NSVZ
      IF (THETAK(59).NE.0.) IL = K
      IF (IB.LE.(IL+NSVZ)) GO TO 100
      RK(2) = IL                                                    AUT00700
      RK(3) = IB
      RK(4) = G(IL-1)                                               AUT00720
      RK(5) = 0.                                                    AUT00730
      RK(6) = THETAK(56)                                            AUT00740
      RK(7) = 0.                                                    AUT00750
      RK(8) = 2.0                                                   AUT00760
   21 IF (THETAK(58).EQ.(-1.)) GO TO 499
C      - -  SOLID AFTER VALOR  - -
   30 RK(13) = THETAK(54)
      RK(14) = -ISR                                                 AUT00620
      RK(15) = G(ISR-1)/THETAK(53)
      RK(16) = RK(15)
      RK(17) = 1.0                                                  AUT00650
      RK(18) = THETAK(53)
      RK(19) = 2.0                                                  AUT00670
   40 RK(1) = 2.                                                    AUT00770
      GO TO 500
C      - -  SOLID ONLY  - -
  100 IF (THETAK(58).EQ.(-1.)) GO TO 1000
      IF (IB+INT(THETAK(53)).GT.150) GO TO 200
      RK(2) = THETAK(54)
      RK(3) =-ISR                                                   AUT00790
```

```
      RK(4)  = G(ISR-1)/THETAK(53)
      RK(5) = RK(4)                                              AUT00810
      RK(6) = 1.
      RK(7)  = THETAK(53)
      RK(8) = 2.0                                                AUT00840
  499 RK(1) = 1.
  500 S16 = -1.                                                  AUT00850
      CALL MP1
  200 S1 = 2.0200                                                AUT00870
      CALL UNCLE                                                 AUT00880
 1000 RETURN                                                     AUT00890
      END                                                        AUT00900
```

```
      SUBROUTINE REZONE                                                  REZ00010
C                                                                        REZ00020
C     COMPILED MAY 16, 1966                                              
C                                                                        REZ00050
C           SUBROUTINE REZONE CALCULATES NEW ZONAL PROPERTIES            REZ00060
C           BETWEEN ANY TWO POINTS (IL,IR), SHIFTS THE PRESENT           REZ00070
C           ARRAYS TO FIT THESE POINTS.                                  REZ00080
C           CALLED FROM EITHER  1MAIN OR  AUTOZ                          REZ00090
C                                                                       *REZ00100
C*               S P U T T E R   C O M M O N                           **REZ00110
C*                                                                     **REZ00120
C*                                                                     **REZ00130
      COMMON   LMDA(37), NR    , NSMLR , IA    , IB    , ICA   , ICB   ,REZ00140
     1   KMAX  , BLANK1, BLANK2, BLANK3, IAP1  , IBP1  , ICAP1 , ICBP1 ,REZ00150
     2   II    , IG    , NRAD  , BLANK4, IAM1  , IBM1  , ICAM1 , ICBM1 ,REZ00160
     3   IIP1  , IGM1  , IALPHA, BLANK5, TH    , TMAX  , BLANK6, DELPRT,REZ00170
     4   FREQ  , CNTMAX, AR    , ASMLR , PUSHA , PUSHB , BOILA , BOILB ,REZ00180
     5   CVA   , CVB   , SLUG  , ALPHA , HVA   , HVB   , HCA   , HCB   ,REZ00190
     6   EMINA , EMINB , CA    , CB    , GA    , GB    , GL    , GR    ,REZ00200
     7   RHOL  , RHOR  , EPIO  , EPSI  , RIA   , RIB   , RDIA  , RDIB  ,REZ00210
     8   RPIA  , RPIB  , RPDIA , RPDIB , TPRINT, TA    , TB    , TC     REZ00220
      COMMON     TD    , TE    , DTH2  , DTH2P , DTH1  , DTRMIN, DTMAX ,REZ00230
     1   DTMAX1, DTMAX2, DTMAX3, DTR   , SWITCH, CO    , CMIN  , DELTA ,REZ00240
     2   GAMA  , WCRIT , SIGMAQ, AC    , AC03T4, CNVRT , SUMRA , SUMRB ,REZ00250
     3   ROIA  , ROIAM1, ROIB  , ROIBP1, GMS   , S1    , S2    , S3    ,REZ00260
     4   S4    , S5    , S6    , S7    , S8    , S9    , S10   , S11   ,REZ00270
     5   S12   , S13   , S14   , S15   , S16   , S17   , S18   , S19   ,REZ00280
     6   S20   , EO    , FO    , TAU   , ZERO  , R   (152), DELTAR(152),REZ00290
     7  ASQ   (152), RD    (152), VD    (152), RDD   (152), SMLR  (152),REZ00300
     8  DELR  ( 37), P     (152), P1    (152), PB    (152), PB1   (152) REZ00310
      COMMON          P2    (152), SV    (152), RHO   (152), THETA (152),REZ00320
     1  W     (152), E     (152), EI    (152), EK    (152), A     (152),REZ00330
     2  V     (152), G     (152), D     (152), C     (152), X2    (152),REZ00340
     3  X3    (152), X4    (152), X5    (152), X6    (152), X7    (152),REZ00350
     4  SMLA  (152), SMLB  (152), SMLC  (152), SMLD  (152), SMLE  (152),REZ00360
     5  EC    (152), ER    (152), SMLQ  (152), SMLH  (152), BIGA  (152),REZ00370
     6  BIGB  (152), CV    (152), BC    (152), BR    (152), CHIC  (152),REZ00380
     7  CHIR  (152), CAPAC (152), CAPAR (152), CRTC  (152), CRTR  (152),REZ00390
     8  CRTPC (152), GOFR  (152), FEW   (152), CAR   (152), OKLM  ( 37) REZ00400
      COMMON          TELM  ( 37), EKLM  ( 37), ELM   ( 37), FCLM  ( 37),REZ00410
     1  FRLM  ( 37), WLM   ( 37), QLM   ( 37), AMASNO( 37), CHRNO ( 37),REZ00420
     2  ZP1   ( 37), ZP2   ( 37), SOLID ( 37), ECHCK ( 37), RK    (104),REZ00430
     3  RL    ( 37), RHOK  (104), RDK   (104), THETAK(104), TEMP  ( 16),REZ00440
     4  HEAD  ( 12), MAXL      , MAXLM                                   REZ00450
C                                                                        REZ00460
C*                                                                     **REZ00470
C*********************************************************************REZ00480
      COMMON /NAMEC/      RLF   , RINC  , RPUSH , RR    , DR           REZ00490
      COMMON /NAMEC/      KCNT  , LCNT  , MCNT  , N     , INO   , IL, IR REZ00500
      COMMON /NAMEC/ VAL    ( 10), RTABRO(150), RTABTH(150), RTABRD(150),REZ00510
     1  RHOTAB(150), THETAB(150), RDTAB (150)                            REZ00520
      COMMON /PROSWT/ IPROSW                                           REZ00530
      COMMON  /LMSB/ U(1)  /LMSC/ M(1)
C                                                                        PEZ00560
      DIMENSION  XZONE(11,9)                                           REZ00610
      DIMENSION ZN(6)                                                  REZ00600
      DIMENSION  ZK(5), Z(1), WT(1)
      EQUIVALENCE  (M(1),KMX), (M(2),Z(1)), (M(3),WT(1))
      EQUIVALENCE (RK(2),XZONE)                                        REZ00570
      REAL  1K
      LOGICAL IBOIL                                                    REZ00620
C                                                                      REZ00630
C                                                                      REZ00640
C              SET THE NUMBER OF REGIONS TO BE CHANGED                 REZ00650
C                                                                      REZ00660
      NCARD = RK(1)                                                    REZ00670
```

```
      DO 140 J = 1,NCARD                                                REZ00680
C                                                                       REZ00690
C               INITIALIZE VARIBLES THAT KEEP TAB ON AREAS TO BE REZONEDREZ00700
C                                                                       REZ00710
      IRN = 0                                                           REZ00720
      IRO = 0                                                           REZ00730
      IGO = IG                                                          REZ00740
      IL = XZONE(1,J)                                                   REZ00750
      IR = XZONE(2,J)                                                   REZ00760
C                                                                       REZ00770
C               DO WE TURN ON THE PROFILE SWITCH                        REZ00780
C                                                                       REZ00790
      IPROSW = 0                                                        REZ00800
      IF(IR .LT. 0) IPROSW = 1                                          REZ00810
      IR = IABS(IR)                                                     REZ00820
      SAVRV6 = R(IR)                                                    REZ00840
C                                                                       REZ00850
C               SAVE THE APPROPRIATE ARRAYS                             REZ00860
C                                                                       REZ00870
      N = 1                                                             REZ00890
C                                                                       REZ00900
      DO 99 I = IL,IR                                                   REZ00910
C                                                                       REZ00920
C               SAVE TABLES                                             REZ00930
C                                                                       REZ00940
      RDTAB(N) = RD(I)
      RTABRD(N) = R(I)
      IF (I.EQ.IR) GO TO 99
C                                                                       REZ00970
      RHOTAB(N) = 1./SV(I)                                              REZ00980
      RTABRO(N) = R(I)                                                  REZ00990
C                                                                       REZ01000
      THETAB(N) = E(I)                                                  REZ01010
      RTABTH(N) = R(I)                                                  REZ01020
      CAR(N)=FEW(I)                                                     REZ01030
      CAPAC(N)=X6(I)                                                    REZ01040
      CHIC(N)=BC(I)                                                     REZ01050
      CHIR(N)=BR(I)                                                     REZ01060
      X4(N)=CRTC(I)                                                     REZ01070
      X5(N)=RHO(I)                                                      REZ01080
      EC(N)=G(I)                                                        REZ01090
      IF (EC(N).LE.0.) GO TO 110
      PB(N)=RDD(I)                                                      REZ01100
      SMLQ(N)=SMLR(I)**2                                                REZ01110
C                                                                       REZ01120
      N = N + 1                                                         REZ01140
      IF (N + 1.GT.150) GO TO 110                                       REZ01150
   99 CONTINUE                                                          REZ01160
C                                                                       REZ01170
      KCNT = IR - IL
      LCNT = KCNT                                                       REZ01190
      MCNT = KCNT + 1
C                                                                       REZ01210
      IF(PUSHA) 104,106,105                                             REZ01220
  104 RLF = RD(IG-1)                                                    REZ01230
      GO TO 111                                                         REZ01240
  105 RLF = RD(1)                                                       REZ01250
  106 GO TO 111                                                         REZ01260
C                                                                       REZ01270
  110 S1 = 55.0110                                                      REZ01280
      CALL UNCLE                                                        REZ01290
C                                                                       REZ01300
  111 IBOIL = .FALSE.                                                   REZ01310
      IF(PUSHA) 112,115,113                                             REZ01320
  112 RPUSH = R(IB)                                                     REZ01330
```

```
      IF(IL.GT.IB .OR. IR.LT.IB .OR. BOILA.EQ.0.) GO TO 115          REZ01340
      IBB = IB                                                       REZ01350
      JB1 = IBP1                                                     REZ01360
      X = -1.                                                        REZ01370
      GO TO 114                                                      REZ01380
  113 RPUSH = R(IA)                                                  REZ01390
      IF(IL.GT.IAM1.OR.IR.LT.IAM1 .OR. BOILA.EQ.0.) GO TO 115        REZ01400
      IBB = IAM1                                                     REZ01410
      JB1 = IAM1                                                     REZ01420
      X = 1.                                                         REZ01430
  114 IBOIL = .TRUE.                                                 REZ01440
      RPUSH=R(JB1)+X*(1.-SOLID(24))*G(IBB)*SOLID(17)/A(JB1)          REZ01450
  115 CONTINUE                                                       REZ01460
C                                                                    REZ01470
C               ERROR CHECK                                          REZ01480
C                                                                    REZ01490
      IF((IR-IL) .GT. 149) GO TO 100                                 REZ01500
C                                                                    REZ01510
C               GENERATE  VAL(1-5)                                   REZ01520
C                                                                    REZ01530
      VAL(1) = 0.                                                    REZ01540
      DO 120 I = 2,10                                                REZ01550
      VAL(I) = XZONE(I+1,J)                                          REZ01560
  120 CONTINUE                                                       REZ01570
      IRM1 = IR - 1                                                  REZ01580
      IF(VAL(6) .EQ. 2.) GO TO 801                                   REZ01590
      IF(VAL(6) . NE. 1.) GO TO 100                                  REZ01600
      DO 800 I = IL,IRM1                                             REZ01610
  800 VAL(1) = VAL(1) - DELTAR(I)                                    REZ01620
      GO TO 803                                                      REZ01630
  801 DO 802 I = IL,IRM1                                             REZ01640
  802 VAL(1) = VAL(1) + G(I)                                         REZ01650
  803 CONTINUE                                                       REZ01660
      RR = R(IL)                                                     REZ01670
      CALL GENRAT                                                    REZ01680
C                                                                    REZ01690
  129 CONTINUE                                                       REZ01700
      RNR = R(NR)                                                    REZ01710
      RICA = R(ICA)                                                  REZ01720
      RICB = R(ICB)                                                  REZ01730
C                                                                    REZ01740
C               SHIFT ARRAYS FOR NEW CONDITIONS                      REZ01750
C                                                                    REZ01760
      CALL MOVE                                                      REZ01770
C                                                                    REZ01780
C                                                                    REZ01790
C               ADJUST BOUNDARIES                                    REZ01800
      IRO = IR                                                       REZ01810
      IRN = IL + IFIX(VAL(5))                                        REZ01820
      IR  = IRN                                                      REZ01830
      ITEM = IRN-IRO                                                 REZ01840
      IRM1 = IR - 1
      R(IR) = SAVRV6
C                                                                    REZ01850
C               CALCULATE ZONE QUANTITIES                            REZ01860
C                                                                    REZ01870
      CALL ZONE                                                      REZ01880
C                                                                    REZ01890
C     CALCULATE AREA                                                 REZ01900
C                                                                    REZ01910
      DO 1294 I = IL,IRM1                                            REZ01920
      IF(S10) 1291,1292,1293                                         REZ01930
 1291 A(I) = 1.                                                      REZ01940
      GO TO 1294                                                     REZ01950
 1292 A(I) = 2. * R(I)                                               REZ01960
```

```
      GO TO 1294                                                        REZ01970
 1293 A(I) = 3. * R(I)**2                                               REZ01980
 1294 CONTINUE                                                          REZ01990
C                                                                       REZ02000
      IF (J.EQ.NCARD) GO TO 1300
      JP1 = J + 1
      DO 130 I = JP1,NCARD
      IF (XZONE(1,J).GT.XZONE(2,I)) GO TO 130
      XZONE(1,I) = XZONE(1,I) + FLOAT(ITEM)                             REZ02030
      XZONE(2,I) = XZONE(2,I) + FLOAT(ITEM) * SIGN(1.,XZONE(2,I))       REZ02040
  130 CONTINUE                                                          REZ02050
 1300 CONTINUE
C                                                                       REZ02060
      IF(PUSHA) 136,    137,  131                                       REZ02070
  131 CONTINUE                                                          REZ02080
      IB = IB+ITEM                                                      REZ02090
      IG = IG+ITEM                                                      REZ02100
      ICB = IB                                                          REZ02110
      IF(IA.GT.IL) IA = ISAV1(IG,IA,RPUSH,R(1))                         REZ02120
      IF(ICA.GT.IL) ICA = ISAV1(IG,-ICA,RICA,R(1))                      REZ02130
      NR = ISAV1(IG,0,RNR,R(1))
      GO TO 139                                                         REZ02150
C                                                                       REZ02160
  136 CONTINUE                                                          REZ02170
      IG=IG+ITEM                                                        REZ02180
      IF(IB.GT.IL) IB = ISAV1(IG,-IB,RPUSH,R(1))                        REZ02190
      ICB=ISAV1(IG,ICB,RICB,R(1))                                       REZ02200
      NR = ISAV1(IG,0,RNR,R(1))
      GO TO 139                                                         REZ02220
C                                                                       REZ02230
  137 CONTINUE                                                          REZ02240
      IG = IG+ITEM                                                      REZ02250
      IB = IB + ITEM                                                    REZ02260
      ICB = ISAV1(IG,ICB,RICB,R(1))                                     REZ02270
      IF(ICA.GT.IL) ICA = ISAV1(IG,-ICA,RICA,R(1))                      REZ02280
      NR = ISAV1(IG,0,RNR,R(1))
  139 CONTINUE                                                          REZ02300
C                                                                       REZ02310
      IAP1 = IA+1                                                       REZ02320
      IAM1 = IA-1                                                       REZ02330
      IBP1 = IB+1                                                       REZ02340
      IBM1 = IB-1                                                       REZ02350
      ICAM1 = ICA-1                                                     REZ02360
      ICAP1 = ICA+1                                                     REZ02370
      ICBM1 = ICB - 1                                                   REZ02380
      ICBP1 = ICB + 1                                                   REZ02390
      IGM1 = IG-1                                                       REZ02400
C                                                                       REZ02410
C     UPDATE VAPOR FRACTION OF BOILING ZONE                             REZ02420
C                                                                       REZ02430
      IF(.NOT.IBOIL) GO TO 1398                                         REZ02440
      IF(PUSHA) 1392,1398,1394                                          REZ02450
 1392 IBB = IB                                                          REZ02460
      JB1 = IBP1                                                        REZ02470
      GO TO 1396                                                        REZ02480
 1394 IBB = IAM1                                                        REZ02490
      JB1 = IAM1                                                        REZ02500
 1396 SOLID(24)=1.-(RPUSH-R(JB1))*A(JB1)*X/(G(IBB)*SOLID(17))           REZ02510
 1398 CONTINUE                                                          REZ02520
C                                                                       REZ02530
C                CALCULATE NEW VELOCITYS                                REZ02540
C                                                                       REZ02550
      DO 200 I=IL,IR                                                    REZ02560
      RDOT=R(I)                                                         REZ02570
      LASTN=3                                                           REZ02580
```

```
      CALL INTERP(ROOT,LASTN)                                          REZ02590
      RD(I)=ROOT                                                       REZ02600
  200 CONTINUE                                                         REZ02610
C                                                                      REZ02620
C                  BALANCE THE NEW ENERGIES                            REZ02630
C                                                                      REZ02640
C     REGENERATE ZONE ENERGY, MASS DEPLETION AND NONEQ VARIABLES       REZ02650
      K=1                                                              REZ02660
      SM1=EC(1)                                                        REZ02670
      I=IL                                                             REZ02680
      SM2=G(IL)                                                        REZ02690
  300 EN=0.                                                            REZ02700
      IF(SUMRA.EQ.0.) GO TO 305                                        REZ02710
      RN=0.                                                            REZ02720
      SN=0.                                                            REZ02730
  305 IF(ZP1(26).EQ.0.) GO TO 315                                      REZ02740
      DO 310 JJ = 1,6                                                  REZ02750
      ZN(JJ) = 0.                                                      REZ02760
  310 CONTINUE                                                         REZ02770
  315 SM=AMIN1(SM1,SM2)                                                REZ02780
      EN=EN+THETAB(K)*SM                                               REZ02790
      IF(SUMRA.EQ.0.) GO TO 320                                        REZ02800
      RN=RN+PB(K)*SM/EC(K)                                             REZ02810
      SN=SN+SMLQ(K)*SM                                                 REZ02820
  320 IF(ZP1(26).EQ.0.)GO TO 325                                       REZ02830
      ZN(1)=ZN(1)+CAR(K)*SM                                            REZ02840
      IF (PUSHA.GE.0..AND.RTABRO(K).LT.RNR) GO TO 325
      IF (PUSHA.LT.0..AND.RTABRO(K).GE.RNR) GO TO 325
      ZN(2)=ZN(2)+CAPAC(K)*SM                                          REZ02850
      ZN(3)=ZN(3)+CHIC(K)*SM                                           REZ02860
      ZN(4)=ZN(4)+CHIR(K)*SM                                           REZ02870
      ZN(5)=ZN(5)+X4(K)*SM                                             REZ02880
      ZN(6)=ZN(6)+X5(K)*SM                                             REZ02890
  325 SM1=SM1-SM                                                       REZ02900
      SM2=SM2-SM                                                       REZ02910
      IF(SM1.GT.0.)GO TO 330                                           REZ02920
      IF (K.LT.KCNT) K = K + 1
      SM1=EC(K)                                                        REZ02940
  330 IF(SM2.GT.0.)GO TO 315                                           REZ02950
      E(I)=EN/G(I)                                                     REZ02960
      EI(I)=E(I)                                                       REZ02970
      IF(SUMRA.EQ.0.) GO TO 335                                        REZ02980
      RDD(I)=RN                                                        REZ02990
      SMLR(I)=SQRT(ABS(SN/G(I)))                                       REZ03000
  335 IF(ZP1(26).EQ.0.) GO TO 340                                      REZ03010
      IF (PUSHA.GE.0..AND.I.LT.NR) GO TO 340
      IF (PUSHA.LT.0..AND.I.GT.NR-1) GO TO 340
      FEW(I)=ZN(1)/G(I)                                                REZ03020
      X6(I)=ZN(2)/G(I)                                                 REZ03030
      BC(I)=ZN(3)/G(I)                                                 REZ03040
      BR(I)=ZN(4)/G(I)                                                 REZ03050
      CRTC(I)=ZN(5)/G(I)                                               REZ03060
      RHO(I)=ZN(6)/G(I)                                                REZ03070
      SUMZ=X6(I)*WT(1)
      IF(KMX.EQ.1) GO TO 337
      SUMZ=SUMZ + BC(I)*WT(11)
      IF(KMX.EQ.2) GO TO 337
      SUMZ=SUMZ + BR(I)*WT(21)
      IF(KMX.EQ.3) GO TO 337
      SUMZ=SUMZ + CRTC(I)*WT(31)
      IF(KMX.EQ.4) GO TO 337
      SUMZ=SUMZ + RHO(I)*WT(41)
  337 IF (SUMZ.EQ.0.) GO TO 340
      X6(I) = X6(I) * FEW(I) / SUMZ
      IF (KMX.EQ.1) GO TO 338
```

```
      BC(I) = BC(I) * FEW(I) / SUMZ
      IF (KMX.EQ.2) GO TO 338
      BR(I) = BR(I) * FEW(I) / SUMZ
      IF (KMX.EQ.3) GO TO 338
      CRTC(I) = CRTC(I) * FEW(I) / SUMZ
      IF (KMX.EQ.4) GO TO 338
      RHO(I) = RHO(I) * FEW(I) / SUMZ
  338 CONTINUE
  340 I=I+1                                                    REZ03120
      SM2=G(I)                                                 REZ03130
      IF(I.LT.IR) GO TO 300                                    REZ03140
      CALL UP37A(IRO,IGO)                                      REZ03460
C                                                              REZ03150
C     CALCULATE ZONE TEMPERATURES                              REZ03160
C                                                              REZ03170
      IF(IPROSW.EQ.1) CALL PROFL                               REZ03180
      DO 380 I=IL,IRM1                                         REZ03190
      IF (ZP1(26).EQ.0.) GO TO 342
      IF (PUSHA.GE.0..AND.I.GE.NR) GO TO 364
      IF (PUSHA.LT.0..AND.I.LT.NR) GO TO 364
  342 IC = 0
      TL=0.                                                    REZ03210
      TR=2.*THETA(I-1)                                         REZ03220
      IF(I.EQ.IL)TR=2.*THETA(I)                                REZ03230
  345 THETA(I)=.5*(TL+TR)                                      REZ03240
      SMLA(I) = THETA(I)
      CALL EOS(I)                                              REZ03250
      IF(IC.GT.30)GO TO 365                                    REZ03260
      DE=E(I)-EI(I)                                            REZ03270
      IF(ABS(DE/EI(I)).LE.S19) GO TO 370                       REZ03280
      IF(DE.GT.0.) GO TO 350                                   REZ03290
      IF(IC.GT.0) GO TO 355                                    REZ03300
      TL=THETA(I)                                              REZ03310
      TR=TL*2.                                                 REZ03320
      GO TO 345                                                REZ03330
  350 TR=THETA(I)                                              REZ03340
      GO TO 360                                                REZ03350
  355 TL=THETA(I)                                              REZ03360
  360 IC=IC+1                                                  REZ03370
      GO TO 345                                                REZ03380
  365 WRITE(6,1000) I                                          REZ03390
 1000 FORMAT(29H0ENERGY DISCONTINUOUS IN ZONE,I3)              REZ03400
      GO TO 380                                                REZ03410
  364 ZK(1) = X6(I)
      ZK(2) = BC(I)
      ZK(3) = BR(I)
      ZK(4) = CRTC(I)
      ZK(5) = RHO(I)
      EION = 0.
      DO 368  K = 1,KMX
      KK = 10*K - 9
      LMAX = Z(KK)
      IK = AINT(ZK(K))
      L = INT(IK)
      NV = M(KK+3) + 1
      IF (L.EQ.0) GO TO 367
      DO 366  JJ = 1,L
      NV = NV + 1
      EION = EION + WT(KK) * U(NV)
  366 CONTINUE
      IF (L.EQ.LMAX) GO TO 368
  367 EION = EION + WT(KK)*U(NV+1)*(ZK(K)-IK)
  368 CONTINUE
  369 AMASNI = AMASN(I)
      CV(I) = 1.5 * AMASNI * (1.0 + FEW(I))
```

```
      THETA(I) = (E(I) - AMASNI*EION - EZERO(I)) / CV(I)                02/23/66
      P1(I) = AMASNI * (1.0 + FEW(I)) * THETA(I) / SV(I)
  370 EI(I)=E(I)                                                        REZ03420
  380 CONTINUE                                                          REZ03430
C                                                                       REZ03440
C                                                                       REZ03450
C                                                                       REZ03470
  140 CONTINUE                                                          REZ03480
      RETURN                                                            REZ03490
C                                                                       REZ03500
C               ERROR                                                   REZ03510
  100 S1 = 55.0100                                                      REZ03520
      CALL UNCLE                                                        REZ03530
C                                                                       REZ03540
      END                                                               REZ03550
```

```
      SUBROUTINE CONVTR
C     COMPILED JUNE 6, 1966                                              CONV0010
C*****************************************************************************CONV0020
C                                                                               CONV0030
C*                    S P U T T E R   C O M M O N                              *CONV0040
C*                                                                            **CONV0050
C*                                                                            **CONV0060
      COMMON  LMDA(9405)
      DIMENSION  EC(152)
      EQUIVALENCE  (S1, LMDA(136)),  (EC(1), LMDA(5517))
C                                                                               CONV0080
C*                                                                            **CONV0090
C*****************************************************************************CONV0100
      COMMON /NAMEC/      RLF    , RINC   , RPUSH , RR     , DR               CONV0110
      COMMON /NAMEC/      KCNT   , LCNT   , MCNT  , N      , INO    , IL, IR  CONV0120
      COMMON /NAMEC/ VAL    ( 10), RTABRO(150), RTABTH(150), RTABPO(150),     CONV0130
     1  RHOTAB(150), THETAB(150), RDTAB (150)                                  CONV0140
C                                                                               CONV0150
      COMMON /SREZS/ IZSWT
      DOUBLE PRECISION R0,R1,R2,R3,G1,G2,V5,ZMS
C                                                                               CONV0180
C                M   M1   MN   (SPACE COORDINATES GIVEN)                      CONV0190
C                X   X    X    SPECIAL CASE   (X MEANS VALUE GIVEN)           CONV0200
C                X   X    0                                                   CONV0210
C                X   0    X    SPECIAL CASE                                   CONV0220
C                0   X    X    ERROR                                          CONV0230
C                0   0    X    ERROR                                          CONV0240
C                0   X    0                                                   CONV0250
C                X   0    0                                                   CONV0260
C                                                                            . CONV0270
C              MASS - SPACE SWITCH                                            CONV0280
      IV6 = VAL(6) + 0.1                                                      CONV0290
C                                                                             CONV0300
      ITEM = 0                                                                CONV0310
      TEM = ABS(VAL(1))-ABS(VAL(3))                                           CONV0320
      DO 190 I = 1,3                                                          CONV0330
      IF((VAL(6)-1.5)*VAL(I).GE.0.) GO TO 190
      R0=DBLE(RR)
      K=MIN0(KCNT,2)
   80 IF(RTABRO(K).GT.RR.OR.K.EQ.KCNT) GO TO 90
      K=K+1
      GO TO 80
   90 CONTINUE
      GO TO (110,150),IV6
C
C                SPACE COORDINATES                                            CONV0350
C                                                                             CONV0360
  110 CONTINUE                                                                CONV0370
C                                                                             CONV0410
C                CHECK FOR MIS-SPECIFICATION                                  CONV0420
C                                                                             CONV0430
                                          ERROR                               CONV0440
      IF(VAL(3).GT.0.0 .AND. VAL(1).EQ.0.0) GO TO 180                         CONV0450
C                                                                             CONV0460
C                CHECK FOR SPECIAL CASE                                       CONV0470
C                                                                             CONV0480
      IF(I.EQ.3 .AND. (VAL(1).GT.0.0 .AND. VAL(3).GT.0.0)) GO TO 130          CONV0490
  120 CONTINUE                                                                CONV0500
      R1=R0
      IF(KCNT.GT.2) GO TO 122
      G1 = 1.D20
      GO TO 124
  122 IF(SNGL(R1).EQ.RTABRO(K-1).AND.IZSWT.EQ.(-1)) GO TO 123
      R2=DBLE(RTABRO(K))
      G1=ZMS(R1,R2,K)
      GO TO 124
  123 G1=DBLE(EC(K-1))
  124 G2=DBLE(VAL(I))
      IF(G2.GE.G1) GO TO 126
```

```
  125 CALL ZRAD(R1,R3,G2,VS,K)
      GO TO 128
  126 IF(G2.GT.G1) GO TO 127
      R3=R2
      GO TO 128
  127 R1=R2
      G2=G2-G1
      IF(K.EQ.KCNT) GO TO 125
      K=K+1
      GO TO 122
  128 VAL(I)=SNGL(R3-R0)
      GO TO 190
C                                                                    CONV0590
  130 CONTINUE                                                       CONV0600
      VAL(3) = TEM                                                   CONV0610
      ITEM = 1                                                       CONV0620
      GO TO 120                                                      CONV0630
C                                                                    CONV0640
C               MASS COORDINATES                                     CONV0650
C                                                                    CONV0660
  150 CONTINUE                                                       CONV0710
C                                                                    CONV0720
C               CHECK FOR MIS-SPECIFICATION                          CONV0730
C                                                 ERROR              CONV0740
      IF(VAL(3).LT.0.0 .AND. VAL(1).EQ.0.0) GO TO 180                CONV0750
C                                                                    CONV0760
C               CHECK FOR SPECIAL CASE                               CONV0770
C                                                                    CONV0780
      IF(I.EQ.3 .AND. (VAL(1).LT.0.0 .AND. VAL(3).LT.0.0)) GO TO 17  CONV0790
C                                                                    CONV0800
  160 G1 = 0.D0
      R3=R0+DBLE(ABS(VAL(I)))
      R1=R0
      IF(KCNT.GE.2) GO TO 162
  161 G1=ZMS(R1,R3,K)+G1
      GO TO 165
  162 R2=DBLE(RTABRO(K))
      IF(R2.GT.R3) GO TO 161
      IF(SNGL(R1).EQ.RTABRO(K-1).AND.IZSWT.EQ.(-1)) GO TO 163
      G1=ZMS(R1,R2,K)+G1
      GO TO 164
  163 G1=DBLE(EC(K-1))+G1
  164 IF(R2.EQ.R3) GO TO 165
      R1=R2
      IF(K.EQ.KCNT) GO TO 161
      K=K+1
      GO TO 162
  165 VAL(I) = -SNGL(G1)
      GO TO 190
C                                                                    CONV0870
  170 CONTINUE                                                       CONV0880
      VAL(I) = TEM                                                   CONV0890
      ITEM = 1                                                       CONV0900
      GO TO 160                                                      CONV0910
C                                                                    CONV0920
  180 CONTINUE                                                       CONV0930
  181 S1 = 43.0181                                                   CONV0940
      WRITE (6,10)                                                   CONV0950
      RETURN                                                         CONV0960
C                                                                    CONV0970
  190 CONTINUE                                                       CONV0980
      IF(ITEM .EQ. 0) RETURN                                         CONV0990
      VAL(3) = ABS(VAL(1)) - ABS(VAL(3))                             CONV1000
      RETURN                                                         CONV1010
C                                                                    CONV1020
   10 FORMAT ( 52H1A MIS-SPECIFICATION HAS BEEN FOUND IN THE ZONE DATA) CONV1030
      END                                                            CONV1040
```

```
      SUBROUTINE ZONE
C     COMPILED JULY 15, 1966
C*****************************************************************
C
C*                    S P U T T E R   C O M M O N                 *
C*                                                                *
C*                                                                *
      COMMON   LMDA(37), NR    , NSMER , IA    , IB    , ICA   , ICB   ,
     1   KMAX  , BLANK1, BLANK2, BLANK3, IAP1  , IBP1  , ICAP1 , ICBP1 ,
     2   II    , IG    , NRAD  , BLANK4, IAM1  , IBM1  , ICAM1 , ICBM1 ,
     3   IIP1  , IGM1  , IALPHA, BLANK5, TH    , TMAX  , BLANK6, DELPRT,
     4   FREQ  , CNTMAX, AR    , ASMER , PUSHA , PUSHB , BOILA , BOILB ,
     5   CVA   , CVB   , SLUG  , ALPHA , HVA   , HVB   , HCA   , HCB   ,
     6   EMINA , EMINB , CA    , CB    , GA    , GB    , GL    , GR    ,
     7   RHOL  , RHOR  , EPIO  , EPSI  , RIA   , RIB   , RDIA  , RDIB  ,
     8   RPIA  , RPIB  , RPDIA , RPDIB , TPRINT, TA    , TB    , TC
      COMMON     TD    , TE    , DTH2  , DTH2P , DTH1  , DTRMIN, DTMAX ,
     1   DTMAX1, DTMAX2, DTMAX3, DTR   , SWITCH, CO    , CMIN  , DELTA ,
     2   GAMA  , WCRIT , SIGMAQ, AC    , ACO3T4, CNVRT , SUMRA , SUMRB ,
     3   ROIA  , ROIAM1, ROIB  , ROIBP1, GMS   , S1    , S2    , S3    ,
     4   S4    , S5    , S6    , S7    , S8    , S9    , S10   , S11   ,
     5   S12   , S13   , S14   , S15   , S16   , S17   , S18   , S19   ,
     6   S20   , EO    , FO    , TAU   , ZERO  , R   (152), DELTAR(152),
     7  ASQ   (152), RU    (152), VD    (152), RDU   (152), SMLR  (152),
     8  DELR  ( 37), P     (152), P1    (152), PB    (152), PB1   (152)
      COMMON          P2    (152), SV    (152), RHO   (152), THETA (152),
     1  W     (152), E     (152), EI    (152), EK    (152), A     (152),
     2  V     (152), G     (152), D     (152), C     (152), X2    (152),
     3  X3    (152), X4    (152), X5    (152), X6    (152), X7    (152),
     4  SMLA  (152), SMLB  (152), SMLC  (152), SMLD  (152), SMLE  (152),
     5  EC    (152), ER    (152), SMLQ  (152), SMLH  (152), BIGA  (152),
     6  BIGB  (152), CV    (152), BC    (152), BR    (152), CHIC  (152),
     7  CHIR  (152), CAPAC (152), CAPAR (152), CRTC  (152), CRTR  (152),
     8  CRTPC (152), GOFR  (152), FEW   (152), CAR   (152), OKLM  ( 37)
      COMMON          TELM  ( 37), EKLM  ( 37), ELM   ( 37), FCLM  ( 37),
     1  FRLM  ( 37), WLM   ( 37), QLM   ( 37), AMASNO( 37), CHRNO ( 37),
     2  ZP1   ( 37), ZP2   ( 37), SOLID ( 37), ECHCK ( 37), RK    (104),
     3  RL    ( 37), RHOK  (104), RDK   (104), THETAK(104), TEMP  ( 16),
     4  HEAD  ( 12), MAXL      , MAXLM
C
C*                                                                *
C*****************************************************************
      COMMON /NAMEC/     RLF   , RINC  , RPUSH , RR    , DR
      COMMON /NAMEC/     KCNT  , LCNT  , MCNT  , N     , INO   , IL, IR
      COMMON /NAMEC/ VAL   ( 10), RTABRO(150), RTABTH(150), RTABRD(150),
     1  RHOTAB(150), THETAB(150), RDTAB (150)
C
      COMMON /SREZS/ IZSWT
      DOUBLE PRECISION R0,R1,R2,R3,R4,G1,G2,GS,VS,VOL,ZMS
C     CALCULATE G(I),R(I),DELTAR(I),SV(I)
      IRM1=IR-1
      IRM2 = IR - 2
      ILP1=IL+1
      K=MINO(KCNT,2)
   80 IF(RTABRO(K).GT.R(IL).OR.K.EQ.KCNT) GO TO 90
      K=K+1
      GO TO 80
   90 CONTINUE
      IF(VAL(6).EQ.2.) GO TO 120
C     SPACE ZONING
C     CALCULATE ZONE WIDTHS
      DELTAR(IL)=VAL(2)
      IF (ILP1.GE.IRM2) GO TO 930
      DO 92 I = ILP1,IRM2
      DELTAR(I)=DELTAR(I-1)*VAL(4)
   92 CONTINUE
      DELTAR(IRM1) = VAL(3)
      DO 93 I = ILP1,IRM2
```

```
      J = ILP1 + IRM2 - I
      WTR = FLOAT(IRM1 - J)/FLOAT(IRM1 - IL)
      DELTAR(J) = WTR*DLLTAR(J) + (1.-WTR)*DELTAR(J+1)/VAL(4)
   93 CONTINUE
C     CALCULATE ZONE BOUNDARIES
  930 R1 = DBLE(R(IL))
      DO 94 I = ILP1,IRM1
      R1 = R1 + DBLE(DLLTAR(I-1))
      R(I) = SNGL(R1)
   94 CONTINUE
C     INITIALIZE LEFT BOUNDARY OF ZONING REGION
      I=IL
      R0=DBLE(R(IL))
  104 R4=DBLE(RTABRO(K))
C     INITIALIZE ZONE I
  105 GS = 0.D0
      R1=R0
      R3=R0+DBLE(DELTAR(I))
C     SUBZONE CALCULATION
  106 R2=R3
      IF(K.NE.KCNT) R2=DMIN1(R3,R4)
      GS=ZMS(R1,R2,K) + GS
      R1=R2
      IF(R2.LT.R4.OR.K.EQ.KCNT) GO TO 108
C     INITIALIZE NEW SECTION OF DENSITY PROFILE
      K=K+1
      R4=DBLE(RTABRO(K))
C     IS ZONE I COMPLETE
  108 IF(R2.LT.R3) GO TO 106
C     ESTABLISH MASS, SPECIFIC VOLUME AND RIGHT BOUNDARY OF ZONE
      G(I)=SNGL(GS)
      VOL=R3-R0
      GO TO (114,111,112),IALPHA
  111 VOL=VOL*(R3+R0)
      GO TO 114
  112 VOL=VOL*(R3**2+R3*R0+R0**2)
  114 SV(I)=SNGL(VOL/GS)
      I=I+1
      IF (I.EQ.IR) GO TO 131
C     GENERATE NEW ZONE
      R0=R3
      GO TO 105
C     MASS ZONING
  120 CONTINUE
C     CALCULATE ZONE MASSES
      G(IL)=VAL(2)
      DO 121 I = ILP1,IRM2
      G(I)=G(I-1)*VAL(4)
  121 CONTINUE
      G(IRM1) = VAL(3)
      DO 122 I = ILP1, IRM2
      J = ILP1 + IRM2 - I
      WTR = FLOAT(IRM1 - J)/FLOAT(IRM1 - IL)
      G(J) = WTR*G(J) + (1.-WTR)*G(J+1)/VAL(4)
  122 CONTINUE
C     INITIALIZE LEFT BOUNDARY OF ZONING REGION
      R3=DBLE(R(IL))
      I=IL
      IF (K.LT.KCNT) GO TO 123
      G1 = 1.D20
      GO TO 125
  123 IF(R(IL).EQ.RTABRO(K-1).AND.IZSWT.EQ.(-1)) GO TO 124
      G1=ZMS(R3,DBLE(RTABRO(K)),K)
      GO TO 125
  124 G1 = DBLE(EC(K-1))
C     INITIALIZE ZONE I
  125 G2=DBLE(G(I))
      R2=R3
```

```
      VOL = 0.D0
C     SUBZONE CALCULATION
  126 GS=DMIN1(G1,G2)
      R1=R2
      CALL ZRAD(R1,R2,GS,VS,K)
      G1=G1-GS
      G2=G2-GS
      VOL=VOL+VS
      IF (G1.GT.0.D0) GO TO 127
      IF (K.LT.KCNT) GO TO 1260
      G1 = G2
      GO TO 127
C     INITIALIZE NEW SECTION OF DENSITY PROFILE
 1260 K = K + 1
      G1=DBLE(EC(K-1))
      IF(IZSWT.EQ.0) G1=ZMS(DBLE(RTABRO(K-1)),DBLE(RTABRO(K)),K)
C     IS ZONE I COMPLETE
  127 IF (G2.GT.0.D0) GO TO 126
C     ESTABLISH WIDTH, SPECIFIC VOLUME AND RIGHT BOUNDARY OF ZONE
      DELTAR(I)=SNGL(R2-R3)
      R3=R2
      SV(I)=SNGL(VOL)/G(I)
      I=I+1
      IF (I.EQ.IR) GO TO 130
      R(I)=SNGL(R3)
      GO TO 125
C     IS RIGHT REGION BOUNDARY PRESCRIBED
  130 IF (R(IR).EQ.1.E20) GO TO 150
  131 R2 = DBLE(R(IR))
      DO 135 I = ILP1,IRM1
      J = ILP1 + IRM1 - I
      WTR = FLOAT(IR-J)/FLOAT(IR-IL)
      R2 = R2 - DBLE(DELTAR(J))
      R(J) = WTR*R(J) + (1.-WTR)*SNGL(R2)
  135 CONTINUE
      GO TO 160
  150 R(IR) = SNGL(R3)
  160 RETURN
      END
```

```
      SUBROUTINE ZRAD(X1,X2,GG,VV,KK)
      COMMON LMDA(9405)
      EQUIVALENCE (S1,LMDA(136)),(IALPHA,LMDA(62))
C*****************************************************  ****************
      COMMON /NAMEC/      RLF    , RINC   , RPUSH , RR       DR
      COMMON /NAMEC/      KCNT   , LCNT   , MCNT  , N        INO    , IL, IR
      COMMON /NAMEC/ VAL    ( 10), RTABRO(150), RTABTH(  0), RTABRD(150),
     1  RHOTAB(150), THETAB(150), RDTAB (150)
C
C     COMPUTE RIGHT BOUNDARY R2 AND VOLUME VS GIVEN LE   BOUNDARY R1
C     AND MASS GS OF A REGION IN WHICH DENSITY IS LINE   IN R**ALPHA
      COMMON /SREZS/IZSWT
      DOUBLE PRECISION R1,R2,GS,VS,C1,C2,X1,X2,GG,VV
      DOUBLE PRECISION  C3, C4, C5, C6
      R1=X1
      GS=GG
      K=KK
      IF(K.GE.2) GO TO 5
      K=2
      GO TO 12
    5 IF (RTABRO(K).NE.RTABRO(K-1)) GO TO 10
      R2=R1
      VS = 0.D0
      GO TO 60
   10 IF (IZSWT.EQ.(-1)) GO TO 12
      S = RHOTAB(K) - RHOTAB(K-1)
      IF(S.NE.0.)GO TO 30
   12 VS = GS/DBLE(RHOTAB(K-1))
      GO TO (15,20,25),IALPHA
   15 R2 = VS + R1
      GO TO 60
   20 R2 = DSQRT(VS + R1**2)
      GO TO 60
   25 R2 = (VS + R1**3)**(1.D0/3.D0)
      GO TO 60
   30 A1 = RTABRO(K-1)
      A2 = RTABRO(K)
      C1 = R1
      GO TO (34,32,33), IALPHA
   32 A1 = A1**2
      A2 = A2**2
      C1 = R1**2
      GO TO 34
   33 A1 = A1**3
      A2 = A2**3
      C1 = R1**3
   34 S = S/(A2 - A1)
      C3 = DBLE(A1 - RHOTAB(K-1)/S)
      C4 = GS*DBLE(2./S)
      C5 = C1 - C3
      C2 = C3
      C6 = C5**2 + C4
      IF (C6.GT.0.D0) C2 = C3 + DSIGN(1.D0,C5)*DSQRT(C
      GO TO (35,36,37), IALPHA
   35 R2 = C2
      GO TO 38
   36 R2 = DSQRT(C2)
      GO TO 38
   37 R2 = C2**(1.D0/3.D0)
   38 VS = C2 - C1
   60 X2=R2
      VV=VS
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION ZMS(R1,R2,KK)
C     COMPILED APRIL 15, 1966
C*****************************************************************
      COMMON /NAMEC/     RLF    , RINC   , RPUSH , RR     , DR
      COMMON /NAMEC/     KCNT   , LCNT   , MCNT  , N      , INO    , IL, IR
      COMMON /NAMEC/ VAL   ( 10), RTABRO(150), RTABTH(150), RTABRD(150),
     1  RHOTAB(150), THETAB(150), RDTAB (150)
C
C     FORMERLY NAMED SAV
C     COMPUTES MASS FOR A REGION WITH DENSITY LINEAR (IZSWT=0) OR
C     CONSTANT (IZSWT=-1) WITH RESPECT TO R**IALPHA
      COMMON LMDA(9405)
      EQUIVALENCE(S1,LMDA(136)),(IALPHA,LMDA(62)),(ALPHA,LMDA(79))
      DOUBLE PRECISION R1,R2,ZMS,B1,B2,R1A,R2A
      COMMON/SREZS/IZSWT
      K=KK
      IF(K.GE.2) GO TO 5
      K=2
    2 S=0.
      GO TO 12
    5 ZMS = 0.D0
      IF (RTABRO(K).EQ.RTABRO(K-1)) RETURN
      IF (IZSWT.EQ.(-1)) GO TO 2
      S = RHOTAB(K) - RHOTAB(K-1)
   12 R1A = R1
      R2A = R2
      A1 = RTABRO(K-1)
      A2 = RTABRO(K)
      GO TO (25,15,20),IALPHA
   15 R1A=R1A**2
      R2A=R2A**2
      A1 = A1**2
      A2 = A2**2
      GO TO 25
   20 R1A=R1A**3
      R2A=R2A**3
      A1 = A1**3
      A2 = A2**3
   25 S = S/(A2-A1)
      B1 = DBLE(RHOTAB(K-1) - S*A1)
      B2 = DBLE(.5 * S)
      ZMS = B1*(R2A-R1A) + B2*(R2A**2 - R1A**2)
      IF(ZMS.GE.0.D0) RETURN
      S1=48.0025
      CALL UNCLE
      END
```

```
      SUBROUTINE PROFL
C     COMPILED MARCH 30, 1966
C*****************************************************************************
C                                                                              **
C                        S P U T T E R   C O M M O N                           **
C*                                                                             **
C*                                                                             **
C*
      COMMON   LMDA(37), NR    , NSMLR , IA    , IB    , ICA   , ICB   ,
     1   KMAX  , BLANK1, BLANK2, BLANK3, IAP1  , IBP1  , ICAP1 , ICBP1 ,
     2   II    , IG    , NRAD  , BLANK4, IAM1  , IBM1  , ICAM1 , ICBM1 ,
     3   IIP1  , IGM1  , IALPHA, BLANK5, TH    , TMAX  , BLANK6, DELPRT,
     4   FREQ  , CNTMAX, AR    , ASMLR , PUSHA , PUSHB , BOILA , BOILB ,
     5   CVA   , CVB   , SLUG  , ALPHA , HVA   , HVB   , HCA   , HCB   ,
     6   EMINA , EMINB , CA    , CB    , GA    , GB    , GL    , GR    ,
     7   RHOL  , RHOR  , EPIO  , EPSI  , RIA   , RIB   , RDIA  , RDIB  ,
     8   RPIA  , RPIB  , RPDIA , RPDIB , TPRINT, TA    , TB    , TC
      COMMON     TD    , T     , DTH2  , DTH2P , DTH1  , DTRMIN, DTMAX ,
     1   DTMAX1, DTMAX2, DTMAX3, DTR   , SWITCH, CO    , CMIN  , DELTA ,
     2   GAMA  , WCRIT , SIGMAQ, AC    , ACO3T4, CNVRT , SUMRA , SUMRB ,
     3   ROIA  , ROIAM1, ROIB  , ROIBP1, GMS   , S1    , S2    , S3    ,
     4   S4    , S5    , S6    , S7    , S8    , S9    , S10   , S11   ,
     5   S12   , S13   , S14   , S15   , S16   , S17   , S18   , S19   ,
     6   S20   , EO    , FO    , TAU   , ZERO  , R   (152), DELTAR(152),
     7   ASQ (152), RD    (152), VD  (152), RDD (152), SMLR  (152),
     8   DELR ( 37), P     (152), P1  (152), PB  (152), PB1   (152)
      COMMON          P2    (152), SV  (152), RHO (152), THETA (152),
     1   W   (152), E     (152), EI  (152), EK  (152), A     (152),
     2   V   (152), G     (152), D   (152), C   (152), X2    (152),
     3   X3  (152), X4    (152), X5  (152), X6  (152), X7    (152),
     4   SMLA(152), SMLB  (152), SMLC(152), SMLD(152), SMLE  (152),
     5   EC  (152), ER    (152), SMLQ(152), SMLH(152), BIGA  (152),
     6   BIGB(152), CV    (152), BC  (152), BR  (152), CHIC  (152),
     7   CHIR(152), CAPAC (152), CAPAR(152), CRTC(152), CRTR  (152),
     8   CRTPC(152), GOFR (152), FEW (152), CAR (152), OKLM  ( 37)
      COMMON          TELM ( 37), EKLM( 37), ELM ( 37), FCLM  ( 37),
     1   FRLM( 37), WLM   ( 37), QLM ( 37), AMASNO( 37), CHRNO ( 37),
     2   ZP1 ( 37), ZP2   ( 37), SOLID( 37), ECHCK ( 37), RK    (104),
     3   RL  ( 37), RHOK  (104), RDK (104), THETAK(104), TEMP  ( 16),
     4   HEAD( 12), MAXL       , MAXLM
C                                                                              **
C*                                                                             **
C*****************************************************************************
      COMMON /NAMEC/     KLF   , RINC  , RPUSH , RR    , DR
      COMMON /NAMEC/     KCNT  , LCNT  , MCNT  , N     , INO   , IL, IR
      COMMON /NAMEC/ VAL   ( 10), RTABRO(150), RTABTH(150), RTABRD(150),
     1  RHOTAB(150), THETAB(150), RDTAB (150)
C     SOLVE QUADRATIC FOR COEFFICIENTS
C
C               E1 = A0 + A1*R1 + A2*R1**2
C               E2 = A0 + A1*R2 + A2*R2**2
C               EGS = RHO*A0*(R2**ALPHA-R1**ALPHA) +
C                     (ALPHA*RHO*A1)/(ALPHA+1.0)*(R2**(ALPHA+1.0)-R1**
C                                                 (ALPHA+1.0)) +
C                (ALPHA*RHO*A2)/(ALPHA+2.0)*(R2**(ALPHA+2.0)-R1**(ALPHA+
C                                                 2.0))
C
C
C
      ZS1(R2,R1) = R2-R1
      ZS2(R2,R1) = R2 + R1
      ZS3(R2,R1) = R2**2 + R1*R2 + R1**2
      ZS4(R2,R1) = R2**3 + R1*R2**2 + R1**2*R2 + R1**3
      ZS5(R2,R1) = R2**4+ R1*R2**3 + R1**2*R2**2 + R1**3*R2 + R1**4
C
      IRM1 = IR-1
      EG = 0.0
      K = IL
```

```
      DO 100 I = IL,IRM1
      IF (E(I).NE.0.) GO TO 20
      K = I + 1
      GO TO 100
   20 EG = EG + E(I)*G(I)
  100 CONTINUE
      IF (K.GE.IR-2) RETURN
      E1 = E(K-1)
      E2 = E(IR)
      R1 = R(K)
      R2 = R(IR)
  101 CONTINUE
      EGS = SV(K) * EG
C
C                CALCULATE COEFFICIENTS
C
      B1 = ZS1(R2,R1)
      GO TO (110,120,130),IALPHA
  110 CONTINUE
C
      B2 = (0.5*ZS2(R2,R1)-R2)
      B3 = (ZS3(R2,R1)/3.0-R2**2)
      A2 = ((EGS/B1-E2)-(E2-E1)*B2/B1)/(-ZS2(R2,R1)*B2+B3)
      GO TO 140
C
  120 CONTINUE
      B2 = (2.0*ZS3(R2,R1)/(3.0*ZS2(R2,R1))-R2)
      B3 = 0.5*ZS4(R2,R1)/ZS2(R2,R1)-R2**2
      A2 = ((EGS/(B1*ZS2(R2,R1))-E2)-(E2-E1)*B2/B1)/(-ZS2(R2,R1)*B2+B3)
      GO TO 140
C
  130 CONTINUE
      B2 = (0.75*ZS4(R2,R1)/ZS3(R2,R1)-R2)
      B3 = (0.60*ZS5(R2,R1)/ZS4(R2,R1)-R2**2)
      A2 = ((EGS/(B1*ZS3(R2,R1))-E2)-(E2-E1)*B2/B1)/(-ZS2(R2,R1)*B2+B3)
C
  140 CONTINUE
      A1 = (E2-E1)/B1-A2*ZS2(R2,R1)
      A0 = E1-A1*R1-A2*R1**2
C
C                FIND THE ENERGY FOR EACH ZONE
C
      DO 190 I = K, IRM1
      R2 = R(I+1)
      R1 = R(I)
C
      GO TO (150,160,170),IALPHA
  150 CONTINUE
C
      E(I) = A0+0.5*A1*ZS2(R2,R1) + A2*ZS3(R2,R1)/3.0
      GO TO 180
C
  160 CONTINUE
      E(I) = A0 + 2.0*A1*ZS3(R2,R1)/3.0 + A2*ZS4(R2,R1)/2.0
      GO TO 180
C
  170 CONTINUE
      E(I) = A0 + 0.75*A1*ZS4(R2,R1) + 0.60*A2*ZS5(R2,R1)
  180 CONTINUE
      EI(I) = E(I)
      ICU = I
      IF (E(I).GE.E1) GO TO 190
      GO TO 199
  190 CONTINUE
      RETURN
```

```
  199 CONTINUE
      DO 200 J=K,ICU
      E(J)=E1
      EI(J)=E(J)
      EG=EG-E(J)*G(J)
  200 CONTINUE
  202 K = ICU + 1
      IF(ICU.EQ.IRM1) RETURN
      R1 = R(K)
      R2 = R(IR)
      GO TO 101
C
      END
```

SECTION II
FORTRAN ANALYSIS ROUTINE

2.1. INTRODUCTION

The FORTRAN Analysis Routine is written entirely in FORTRAN IV for a 65K-UNIVAC 1108 computer. Conversion to other dialects of FORTRAN IV is not expected to present major obstacles. This routine consists of two independent programs. Program One is a variable renaming program; i.e., it provides for replacement of specified variable names by other names in a FORTRAN source deck, edits the deck by providing appropriate spacing, punches out a new sequenced deck, and provides a listing of this deck. In addition, it generates a tape containing all the required information for the other program. Program Two provides a printout of an extensive cross reference dictionary for the variables used in the program and the source line in which each variable appeared, including a description of the variable, its uses in each subprogram, and the subprogram in which it appears. It is intended that these programs be widely used for program maintenance.

2.2. DESCRIPTION

In general, the input to Program One consists of an unlimited number of compilable FORTRAN IV subprograms, each of which is preceded by up to 15 "valid" strings which may appear in the subprograms that follow. For each "valid" string to be substituted for, a substitution string must be provided which maintains the compilability of the newly created statement.

A string is a sequence of characters. Only a valid string may be modified. To be valid, a string must be one of the following:

1. A variable name or a common block name or a subprogram name (hereafter the word "identifier" will be used to refer to one of these three),
2. A variable or subprogram name followed by a balanced set of parentheses,
3. The contents of a balanced set of parentheses,
4. Any string contained between consecutive commas,
5. Any string contained between a left parenthesis and the first comma that follows it, if any, prior to the corresponding right parenthesis,
6. Any string contained between the right-most comma following a left parenthesis, if any, and the corresponding right parenthesis;

and it must not be any of the following:

1. An arithmetic or logical operator,
2. A single non-alphabetic character,
3. .TRUE., .FALSE., .T., .F.,
4. A real or double precision or integer constant,
5. Any hollerith string (as appearing in a format statement or data statement or as an argument of a call).

In an arithmetic statement or arithmetic function statement, the left side of an equal sign is always a valid string; however, the right side may or may not be a valid string. The entire contents of the balanced set of parentheses following an IF statement is not necessarily a valid string.

In determining whether a substitution should take place in the subprograms, blanks are ignored. That is, all statements in the input programs are compressed as though no blanks were left anywhere. Of course, blanks are not deleted from hollerith fields.

2.3. EXAMPLES

1. A=B(1,I)+SQRT(D(2,2))-F(I,J+1,K)+2 *G
2. IF(Z.LT..3) CALL EXIT

In the above two statements, only the following are valid strings:

* A	* F
E(1,I)	* I
* B	J+1
* I	* J
SQRT(D(2,2))	* K
* SQRT	* G
* D(2,2)	* Z
* D	* EXIT
F(I,J+1,K)	

Two passes are normally made through each statement in a program. A statement, including all its continuation cards, is read, and blanks are deleted. During the first pass every valid string is checked for substitution, and the substitution, if any, is made. During the second pass all identifiers and all variables having a completely numerical subscript are written on tape along with the new source line in which each variable appears. Those valid strings listed above with asterisks preceding them would have been recorded on tape. Not only are the name and source line of each identifier and numerically subscripted variable recorded on tape, but the fact that it appeared:

1. In a type statement or specification statement as (1) complex, (2) integer, (3) external, (4) logical, (5) real, (6) double precision, (7) equivalence, (8) parameter, (9) data.
2. On the left side of an equal sign.
3. As an argument of (1) the subprogram itself, (2) a referenced subroutine, (3) a referenced function.
4. As the name of the subprogram in a subroutine statement or function statement.

5. As the name of a referenced subroutine or referenced function.
6. In a statement which specifies a variable to be dimensioned (in a common or type statement with a numerical subscript or in a dimension statement), in which case its dimension and the number of characters in its dimension are also recorded.
7. In a common statement as either (1) a block common name or (2) a variable name, in which case its common block or blank is also recorded.
8. As a variable with a completely numerical subscript, in which case its subscript and the number of characters in its subscript are also recorded.
9. In a read statement.
10. Simply as a variable in none of the above.

This information is recorded at the time of appearance for each and every appearance of every identifier and numerically subscripted variable.

Pass two of Program One edits each statement after substitutions, if any are made, in the following manner: Single blanks are inserted before and after FORTRAN logical operators and the connectors =, +, and -, following a comma and neither before nor after the connectors *, /, **. Blanks do not occur in column 7 except for continuation cards, when they always occur. Identifier names are never split between one card and its continuation card. The continuation card mark in column 6 is always an apostrophe. The following statements are not edited except that the continuation marks are changed to apostrophes and information is placed in columns 73-80: comment cards, FORMAT statements, CONTINUE cards, and END statements. The compound or logical IF statement is split so that the second half of the compound statement begins on a continuation card. On all cards, columns 73-76 will contain the first four characters of the subprogram name appearing as the first name on the FOR card for each subprogram. Columns 77-80 will contain a 4 digit sequence number starting from 1 for each subprogram and increasing by 1.

A new sequenced source deck will be punched. A listing of the source deck will be printed with the additional information that the first line of each statement where a substitution has been made will be so noted. Any statement which required more than 19 continuation cards after editing will be so noted, and the additional cards will be punched and listed. All strings for which substitutions were requested that did not appear in the subprogram just processed will be noted at the end of the listing for that subprogram.

In addition to the pertinent description of Program One given above, the following unrelated facts concerning this program must be stated and understood if it is to be used properly:

1. Statement numbers are in no way affected by the program.
2. This program does not have extensive checking features, and a non-compilable statement may create an endless loop.
3. Each subprogram must begin with a FOR card with column one blank (the 7-8 punch removed) and end with an END statement.
4. Superfluous real, integer, and equivalence statements, created by virtue of variable substitutions, will not be eliminated.
5. No new real- or integer-type statements will be generated to correct for an unintentional change of type when a name substitution changes the type.
6. A name substitution will take place whenever that name appears regardless of whether it is subscripted.
7. A variable plus subscript substitution will take place only when the variable and subscript agree exactly. However, no substitution will take place if the subscript is numeric and appears in a dimension statement or any statement which specifies its dimensionality. Thus, the replacement of R(152) by PCYCLE would not occur in the statement COMMON R(152).
8. The appearance of any variable followed by a left parenthesis not appearing in a specification statement defining its dimensionality will be checked to determine whether it has been

previously dimensioned. If not, it will be assumed to be a function reference. Thus, on substituting C(5) for B, if C has not been previously dimensioned, it will be considered a function and will not be otherwise noted as a possible source error.

9. This program cannot accept comment cards embedded between the first card of a statement and the last continuation card of the statement. Such cards must be removed from a deck before this program is used. Comment cards may appear any place else between a FOR card and an END card of a subprogram.
10. Anything appearing in columns 73-80 on comment cards is deleted, and the sequencing scheme previously described is substituted.
11. The number of + signs, - signs, and logical operators appearing in a compound or logical IF statement must not exceed 20 if this statement is to be reproduced correctly.

Program Two processes the output of Program One. It will print out the following information for each subprogram:

1. The subprogram name plus an 18 character tape identification. This identification is described further in Section 2.4 of this report.
2. An alphabetic listing of all common blocks starting with blank common. For each common block the following information is printed.
   a. A listing of the variables within each block in order of their appearance in the block, and for each variable its octal and decimal origin relative to zero within the block.
   b. An alphabetic listing of the variables within each block, and for each variable its octal and decimal origin relative to zero within the block.

3. An alphabetic listing of the subroutines and functions referenced within this subprogram and the source lines in which they were referenced.

4. An alphabetic listing of each variable appearing in the subprogram. For each variable the following information is printed:

   a. A single line containing the variable name, the block name in which it may appear, its octal and decimal origin relative to zero within that block, its dimensions (if any), the size of its dimensions, and a series of code letters describing its "uses" within the program (i.e., its type, whether it appears anywhere within the subprogram on the left side of an equal sign, whether it appears anywhere within the subprogram as the argument of a call, whether it appears anywhere in the subprogram in a read statement, etc.) A full description and listing of these code letters appears in Section 2.4 of this report and on the first page of the printed output of Program Two.

   b. An ordered listing of every source line in which the variable appears is printed. Following each source line there may appear the same coded letters as are described in (4a), above, if the variable can be modified within that line (by appearing on the left side of an equal sign or as an argument of a function, in a read statement, etc.).

   c. For each variable that is numerically subscripted within the program, a sort order listing for each numeral subscript, followed by a list of all source lines in which it appears with this subscript.

After all subprograms have been processed and each of the above listings has been printed, the program prints an alphabetic listing of all

3. An alphabetic listing of the subroutines and functions referenced within this subprogram and the source lines in which they were referenced.

4. An alphabetic listing of each variable appearing in the subprogram. For each variable the following information is printed:

   a. A single line containing the variable name, the block name in which it may appear, its octal and decimal origin relative to zero within that block, its dimensions (if any), the size of its dimensions, and a series of code letters describing its "uses" within the program (i.e., its type, whether it appears anywhere within the subprogram on the left side of an equal sign, whether it appears anywhere within the subprogram as the argument of a call, whether it appears anywhere in the subprogram in a read statement, etc.) A full description and listing of these code letters appears in Section 2.4 of this report and on the first page of the printed output of Program Two.

   b. An ordered listing of every source line in which the variable appears is printed. Following each source line there may appear the same coded letters as are described in (4a), above, if the variable can be modified within that line (by appearing on the left side of an equal sign or as an argument of a function, in a read statement, etc.).

   c. For each variable that is numerically subscripted within the program, a sort order listing for each numeral subscript, followed by a list of all source lines in which it appears with this subscript.

After all subprograms have been processed and each of the above listings has been printed, the program prints an alphabetic listing of all

identifiers appe ing in any of the subprograms, and for each identifier, an alphabetic listing of the subprograms in wh. i it appeared. The subprograms are listed by subroutine name (not by the name on the FOR card). This ends the output description of Program Two.

Both Program One and Program Two use logical unit 10 to store and process the identifier information. A complete description of the format and information stored on logical unit 10 by Program One is contained in Section 2.5 of this report. Table lengths for Program Two are also given in that section.

2.4. INPUT

It is suggested that FORTRAN code sheets be used for all data input, since the data format has been so designed. There are three types of data input for Program One. Data type 1 is a single card having two fields. It precedes a complete run of Program One which may consist of one or more subprograms. Field 1 - Column 1 (format I1) controls logical unit 10; if field 1 is blank, logical unit 10 will be rewound. This implies either that the unit is blank or that information previously stored on this unit is no longer desired. If field 1 contains a one, it is assumed that logical unit 10 had information previously stored on it from a previous run of Program One; then logical unit 10 will be positioned forward to the end of the previously stored information, ready to receive additional information. Field 1 - Columns 2-6 inclusive are blank. Field 2 - Columns 7-24 (format 3A6) is the secondary identification valid for all subprograms of this run. It may contain any information desired, e.g., the date. It is intended that a program will be written to edit logical unit 10, deleting one version of a subprogram and accepting another version of this same subprogram by utilizing the uniqueness of the 18 character identification in field 2 between runs of Program One.

For Data type 2, one or more cards must precede each subprogram, but no subprogram will require more than 15 of these cards. Fifteen "blocks" have been set aside, internal to Program One, for "substitutions." (A "substitution" is defined as the combination of a source program string to be changed and its replacement string.) A block in which no substitution has ever been placed or one in which a substitution has been deleted without replacement is considered void. A block that is not void is full. To "delete" means to wipe out a substitution. It can only be brought back by fully restating the substitution on a new type 2 card, as described below. A type 2 card may serve any of the following purposes:

1. It may fill a void block with a substitution. The block is then full. The substitution then applies through all subprograms that follow it unless deleted or replaced by another type 2 card.
2. It may delete a particular full block. That block is then void.
3. It may delete all existing full blocks. All blocks will then be void.
4. It may delete a substitution from a full block and replace it with another substitution in the same block. That block is then still full. The new substitution will then apply through all subprograms that follow it unless it is specifically deleted or replaced by still another type 2 card.

Type 2 cards consist of three fields, hereafter referred to as F1, F2, and F3. F1 (Format A6) delineates the last data type 2 card by having some non-blank character in any of columns 1-6 of the data card, (except that column 1 must not have a 7-8 punch). That is, columns 1-6 must be blank for all but the last data type 2 card preceding each subprogram.

F2 (format 71A1) is either entirely blank, in which case it sets the block specified in F3 void, or contains the substitution in columns 7-77, inclusive. If non-blank, the general form of F2 is SOURCE PROGRAM STRING TO BE MODIFIED = ITS REPLACEMENT STRING. The following rules must be observed in the specification of F2.

1. F2 must start in column 7. (It is considered blank if column 7 of F2 is blank.)
2. The two strings must be separated by an equal sign.
3. There must not be any blanks between column 7 and the last character of the replacement string.
4. Column 78 must be blank.

For F3 (Format I2), columns 79 and 80 must contain blanks or the numbers 01-15 or the number 99. F3 may be blank if and only if F1 is non-blank and F2 is blank, i.e., to express the end of type 2 data without supplying any information on F2 and F3. In general, F3 contains the block numbers 01-15 for which an F2 substitution or deletion or replacement substitution is being made. The information for a block need not be ordered, i.e., information for block 07 may be supplied before information for block 01, and blocks may be arbitrarily full or void. If F3 contains a 99, all 15 blocks are void. F2 would be blank if F3 contained a 99, and F1 might or might not be blank. It is perfectly valid for additional type 2 cards to follow an F3 containing a 99 (in which case F1 would be blank when F3 contained a 99).

If all 15 blocks are void upon return from the program that processes the type 2 cards, then Program One will bypass its first pass (the substitution phase) and will presumably run twice as fast. This implies that all one is interested in is the cross reference dictionary generated in pass two. If this is so, then the data type 2 cards preceding each subprogram should consist of a single card with perhaps a single character in column 1.

(However, the single character must not be a 7-8 punch, which transfers control to the 1108 operating system.)

Data type 3 cards are subprogram source cards to be processed. They must start with a FOR card with the column 1 blank (that is the 7-8 punch must be removed). Following column 1 there must be at least one blank preceding and at least one blank following the "FOR" and before its Deck Name. Just prior to the FOR card there must be a data type 2 card with at least one character other than a blank in columns 1-6 (and not a 7-8 punch in column 1). Following the modified FOR card are the source cards, which must be compilable. The source deck must end with an END card.

Program Two requires no card input. A request for logical unit 10 (ASG) must be supplied by the user for both Program One and Program Two.

2.5. APPENDIX: PROGRAM DETAILS

Table I contains information describing the code letters which define the "use" of an identifier (see steps (4a) and (4b) of the Program Two output description in Section 2.2), and the code numbers written on tape by Program One which identify that "use."

Any combination of the code letters appearing in column 1 of Table I may be printed in the Program Two output previously described in step (4a) of Section 2.3, but only those code letters in Table I which have an asterisk preceding them will be printed out as described in step (4b). The latter code letters reference only those statements which can modify a variable.

The information written by Program One on logical unit 10 is written in unformatted I/O (equivalent to binary) and consists of six words per record. The six words make up four fields. Field 1 is in field data and is either (1) the program name appearing on the FOR card, as the first word of the first and last record of each subprogram, or (2) an identifier appear-

Table I

DESCRIPTION OF CODE LETTERS

Code Letter	Code Number(k)	Description
U	- 89	Variable appears in a parameter statement
* D	- 90	Variable appears in a data statement
	- 91	Name appears on FOR Card
Z	- 92	Variable appears in a complex type statement
* X	- 93	Variable appears on left side of an equal sign
* H	- 94	Variable appears in a common block
	- 95	Block name appears in a common statement
* S	- 96	Variable appears as an argument of a call
B	- 97	Variable appears in a double precision type statement
* R	- 98	Variable appears in a read statement
	- 99	Subroutine name on subroutine or function statement card
E	-100	Variable appears in an external type statement
C	-101	Subroutine name appears in a call statement
I	-102	Variable name appears in an integer type statement
* Q	-103	Variable name appears in an equivalence type statement
L	-104	Variable appears in a logical type statement
F	-105	Function name appear in an arithmetic statement or IF statement, or on either side of the equal sign in an arithmetic function statement (but not on a function statement card described in -99 above)
P	-106	Variable appears in a real type statement
A	-107	Variable appears as a program argument on a function or subroutine statement card
* G	-108	Variable appears as the argument of a referenced function as described in -105 above
	$-18 \leq k < 0$	Variable appears in a dimension statement or in a statement describing its dimensionality where $\|k\|$ is the number of characters in its dimension description including parentheses, commas, and digits
	$0 < k \leq 18$	Variable appears in a statement, other than one describing its dimensionality or an equivalence statement, with a numerical subscript where k is the number of characters in its subscript including both parentheses, commas, and digits
	$-218 \leq k < -200$	Variable appears in an equivalence statement with a subscript where $\|k-200\|$ is the number of characters in its subscript, including both parentheses, commas, and digits
	0	Not specified above (as the variables A or B in the arithmetic statement Y=A+B)

ing in the program. Field 1 is left adjusted and filled with blanks on the right if the identifier is less than six characters.

Field 2 is in field data and consists of three words, either:

1. The 18 character identification specified on the data type 1 card of a specific run of Program One. It will appear in the second field of the first record of every subprogram processed during this run.
2. A variable dimension, left adjusted, with no embedded blanks, containing both parentheses and their contents, and filled with blanks on the right. Field 2 will contain this information only if the variable in field 1 is appearing in a statement specifying its dimensionality at the time this record is written.
3. A variable numerical subscript, left adjusted, with no embedded blanks (as described in step 2 above). Field 2 will contain this information only if the variable in field 1 is appearing in any statement other than one specifying its dimensionality at the time this record is written and is numerically subscripted.
4. A block name (which may be blank for blank common). Word 1 of field 2 will contain this information only if the variable is appearing in a common statement at the time the record is written; then words 2 and 3 of field 2 will be blank. If field 1 contains the block name, the first word of field 2 will also contain that block name. All three words of field 2 contain blanks.

Field 3, the fifth word of the record, is the code number appearing in the second column of Table I. The last record of each subprogram contains a -200 in field 3 (generated by the END statement). Field 4 contains the source line number in which the identifier in field 1 appears. Following the last record of the last subprogram is one additional six word record containing a -200 in its fifth word. There is no end-of-file following this.

One can, therefore, determine the end of information by seeking two consecutive records each of which contains -200 in its fifth word.

If a variable appears in a statement specifying its dimensionality or is numerically subscripted, it will appear in two consecutive records. For example, the numerically subscripted variable A(10) in the statement, A(10)=Y appearing in line 17 of a subprogram appears in two records as:

	Field 1	Field 2	Field 3	Field 4
Record n	A	(10)	4	17
Record n+1	A		-93	17
Record n+2	Y		0	17

The purpose of the above description is to provide the user with an opportunity to write his own specialized cross reference dictionary (i.e., his own variation of Program Two. For example, he may wish to print out all variables appearing in READ statements throughout the program in the order in which they are read.

Certain assumptions had to be made in Program Two concerning the internal allocation of space for the many tables required by the cross reference dictionary. This allocation is further complicated by a possible variation in core space that will be available to the user in the near future. It is currently assumed that the total data space must not exceed 40,000. That is, the sum of the "total space" created by the parameter occurrences listed in Table II and fixed working memory must not exceed 40,000. However, as long as the total does not exceed 40,000, any one or more of the five parameters may be modified within the program in the event the individual tables to which they refer are exceeded. This requires the change of a single PARAMETER card in Program Two, which states: PARAMETER Z1=15000, Z2=500, Z3=1000, Z4=2500, Z5=1000, where Z1-Z5 and the number of appearances of Z1-Z5 are as defined in Table II. The "limit" column in Table II refers to the maximum appearances or references or number of such items in any one subprogram.

Table II

DEFINITION OF ENTRIES IN PARAMETER CARD

Parameter	Description	Limit		Number of tables requiring this parameter	=	Total space
Z1	Number of references of variables with or without subscripts	15000	×	1	=	15,000
Z2	Number of unique common block names	500	×	1	=	500
Z3	Number of appearances of variables with unique subscripts	1000	×	6	=	6,000
Z4	Number of appearances of unique identifiers	2500	×	4	=	10,000
Z5	Number of appearances of unique variables that are dimensioned	1000	×	5	=	5,000
						36,500
	Fixed working memory (not subject to change)					3,500
						40,000

SECTION III
DATA-LINK EDITING ROUTINES

3.1. INTRODUCTION

In order to facilitate use of the computer codes developed in the NEIS on the CDC-6600 computer at the Air Force Weapons Laboratory, a data link was installed between the SNEL facility in San Diego and the AFWL facilities in Albuquerque. The link terminals include an IBM-1978 reader-printer at SNEL, which communicates with an IBM-7702 tape terminal at AFWL. The format of the information which can be transmitted or received by the 1978 is unfortunately incompatible with the requirements of the Chippewa operating system for the 6600 computer, so editing procedures are required as an intermediate step in the data transmission process. Two programs, called TOGA and FROGA, were written for the AFWL CDC-160A computer to perform these editing functions.

3.2. TOGA

The TOGA program edits BCD output tapes prepared by the 6600, producing modified tapes which can be read by the 7702, and printed on the 1978. The following editing functions are accomplished:

1. Printer carriage control characters written in FORTRAN form are blank, 0, -, and 1, indicating single space, double space, triple space, and page eject. These actions precede the printing of the line in which they occur. The corresponding 1978 control characters are /, S, T and A, respectively, with the difference that the actions follow the printing of the line in which they occur. The TOGA code therefore replaces each FORTRAN carriage control character by its 1978 counterpart, which is then placed

at the head of the preceding line. If the first character in the line is already a 1978 character, no action is taken; if it is an illegal character, a single space (/) is supplied.

2. The 1978 requires each line to be terminated by a segment mark, which is a special BCD character. This character is placed at the end of each line, after any trailing blanks are deleted.
3. The physical records transmitted over the link are blocked to some extent. The number of lines per block varies from two to seven, depending upon the total number of characters, which may not exceed 328.
4. The length of a line printable on the 1978 is 120 characters, plus the leading control character and the trailing segment mark. If the input lines exceed this length, the overflow goes onto a second line flagged by a leading exclamation point, following which the carriage is double-spaced.
5. Input records less than three characters long are considered noise and are skipped.

3.3. FROGA

The FROGA program edits information transmitted by the 1978 into a form acceptable for job input by the Chippewa system. The following functions are performed:

1. The transmitted records, except possibly the last, consist of four 80 character card images, each followed by a segment mark. These records are read one at a time. For each card image, the segment mark and any trailing blanks are deleted, the latter two at a time.
2. The card images are moved to an output area. The end of each card image is indicated by the following rule: The number of

characters for each card image must be a multiple of 10, and the last two of these must be 6-bit (binary) zeros.

3. As the BCD characters are moved to the output area, they are translated into the Chippewa display code by table lookup.

4. The size of the output buffer is 5120 characters. The buffer is written onto an output "display tape" when full, or when an end-of-record flag is received. The end-of-record flag is an apostrophe in column 1 of a card image. If the buffer is just filled before the end-of-record flag is received (an unlikely occurrence), a special short record consisting of eight binary zeros (48 bits) is written. Either this special short record or the normal short record (less than 512 ten character words) indicates to the Chippewa system that a "logical record" is complete.

5. End-of-file records are written on the display tape when a second apostrophe is received following the first, e.g., in column 2 of the same card image. This end-of-file separates one job from another. A third apostrophe indicates end of transmission, as well as end-of-record and end-of-file. When only one job is transmitted at a time, only single or triple apostrophe flags are used (the preferred procedure). On end-of-transmission, two end-of-file records are written on the display tape.

characters for each card image must be a multiple of 10, and the last two of these must be 6-bit (binary) zeros.

3. As the BCD characters are moved to the output area, they are translated into the Chippewa display code by table lookup.

4. The size of the output buffer is 5120 characters. The buffer is written onto an output "display tape" when full, or when an end-of-record flag is received. The end-of-record flag is an apostrophe in column 1 of a card image. If the buffer is just filled before the end-of-record flag is received (an unlikely occurrence), a special short record consisting of eight binary zeros (48 bits) is written. Either this special short record or the normal short record (less than 512 ten character words) indicates to the Chippewa system that a "logical record" is complete.

5. End-of-file records are written on the display tape when a second apostrophe is received following the first, e.g., in column 2 of the same card image. This end-of-file separates one job from another. A third apostrophe indicates end of transmission, as well as end-of-record and end-of-file. When only one job is transmitted at a time, only single or triple apostrophe flags are used (the preferred procedure). On end-of-transmission, two end-of-file records are written on the display tape.

DISTRIBUTION

No. cys

HEADQUARTERS USAF

1 Hq USAF (AFTAC), Wash, DC 20330

MAJOR AIR COMMANDS

1 AFSC (SCTR), Andrews AFB, Wash, DC 20331

1 AUL, Maxwell AFB, Ala 36112

AFSC ORGANIZATIONS

1 AF Avionics Laboratory, Wright-Patterson AFB, Ohio 45433

1 AF Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio 45433

RTD, Bolling AFB, Wash, DC 20332

1 (RTTW)

1 (RTTG)

1 BSD (BSR), Norton AFB, Calif 92409

1 RADC (EMLAL-1), Griffiss AFB, NY 13442

KIRTLAND AFB ORGANIZATIONS

1 AFSWC (SWEH), Kirtland AFB, NM 87117

AFWL, Kirtland AFB, NM 87117

5 (WLIL)

1 (WLRT)

1 (WLRTH/Capt Whitaker)

2 (WLRTH/Maj Spillman)

OTHER AIR FORCE AGENCIES

Director, USAF Project RAND, via: Air Force Liaison Office, The RAND Corporation, 1700 Main Street, Santa Monica, Calif 90406

1 (Dr. Harold Brode)

1 (RAND Library)

Hq OAR, 1400 Wilson Blvd, Arlington, Va 22209

1 (RROS)

1 (RROP)

AFCRL, L. G. Hanscom Fld, Bedford, Mass 01731

1 (Dr. K.S. Champion)

1 (S. Harowitz)

1 (Dr. W. Pfister)

1 (Dr. Newman)

1 (Dr. Gassman)

DISTRIBUTION (cont'd)

No. cys	
	ARMY ACTIVITIES
1	Commanding Officer, Harry Diamond Laboratories, ATTN: Library, Wash, DC 20438
1	Commanding Officer, US Army Combat Developments Command, Institute of Nuclear Studies, Ft Bliss, 79916
1	Redstone Scientific Information Center, US Army Missile Command, Chief, Document Section, Redstone Arsenal, Ala 35809
	Commanding Officer, Ballistic Research Laboratories, Aberdeen Proving Ground, Md 21005
1	(ATTN: Mr. E.O. Baicy)
1	(Mr. W. Berning)
	Commanding Officer, US Army Electronics Research and Development Laboratory, Ft Monmouth, NJ 07703
1	(Technical Document Center)
1	(Weapon Effects Section)
2	Director, US Army Engineer Research and Development Laboratories, ATTN: STINFO Branch, Ft Belvoir, Va 20260
	NAVY ACTIVITIES
1	Office of the Chief of Naval Operations, Department of the Navy, (OP-75, ATTN: Director, Atomic Energy Division), Wash, DC 20350
1	Chief of Naval Research, Department of the Navy, ATTN: Mr. Winchester, Code 418, Wash, DC 20390
1	Commander, Naval Ordnance Laboratory, ATTN: Dr. Rudlin, White Oak, Silver Spring, Md 20910
1	Director, Special Projects Office, Department of the Navy, Wash, DC 20360
	OTHER DOD ACTIVITIES
2	Director, DASA (Document Library Branch), Wash, DC 20301
3	Commander, Field Command, DASA (FCAG3, Special Weapons Publication Distribution), Sandia Base, NM 87115
3	Director, ARPA, Department of Defense, The Pentagon, Wash, DC 20301
1	Office of Director of Defense Research and Engineering, ATTN: John E. Jackson, Office of Atomic Programs, Rm 3E1071, The Pentagon, Wash, DC 20330
20	DDC (TIAAS), Cameron Station, Alexandria, Va 22314
	AEC ACTIVITIES
1	Sandia Corporation (Information Distribution Division), Box 5800, Sandia Base, NM 87115

DISTRIBUTION (cont'd)

No. cys	
1	Sandia Corporation (Tech Library), P.O. Box 969, Livermore, Calif 94551
	University of California Lawrence Radiation Laboratory, ATTN: Director's Office, P.O. Box 808, Livermore, Calif 94551
1	(Dr. Ronald B. Carr (L-122))
1	(Dr. William Lokke)
	Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87554
1	(Helen Redman, Report Library)
1	(Dr. Robert Thorn, T-Division)

OTHER

1	Massachusetts Institute of Technology, Lincoln Laboratory (Document Library), P.O. Box 73, Lexington, Mass 02173
1	Convair, A Division of General Dynamics Corporation, ATTN: Technical Library, P.O. Box 1950, San Diego, Calif 92112
2	General Atomic Division, General Dynamics Corporation, ATTN: Dr. K. D. Pyatt, 10955 John Jay Hopkins Drive, San Diego, Calif 92121
1	General Electric TEMPO, ATTN: DASA Data Center 735 State Street, Santa Barbara, Calif 93101
1	Stanford Research Institute, ATTN: G-037, External Reports, Menlo Park, Calif 94025
1	Lockheed Missiles and Space Company, Div Lockheed Aircraft Corporation, 1111 Lockheed Way, ATTN: Dr. D. Moffat, Sunnyvale, Calif 94086
1	Kaman Nuclear, Div of Kaman Aircraft Corporation, ATTN: Dr. A.P. Bridges, Garden of the Gods Road, Colorado Springs, Colo 80907
1	Bell Telephone Laboratories, Inc., ATTN: J. W. Foss, Whippany Road, Whippany, NJ 07981
1	Official Record Copy (WLRT/Maj Spillman)

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
General Atomic Division
General Dynamics Corporation
San Diego, California

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE
NUCLEAR EXPLOSION INTERACTION STUDIES
VOLUME III: MISCELLANEOUS CODE DEVELOPMENT

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report 22 July 1965 through 21 July 1966

5. AUTHOR(S) (Last name, first name, initial)
Pyatt, K. D., Jr., et al.

6. REPORT DATE
May 1967

7a. TOTAL NO. OF PAGES
58

7b. NO. OF REFS

8a. CONTRACT OR GRANT NO.
AF29(601)-7035

b. PROJECT NO. 5710

c. Subtask No. 07.002

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
AFWL-TR-66-108, Vol. III

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
Contractor's Report No. GA-7370

10. AVAILABILITY/LIMITATION NOTICES This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of AFWL (WLRT), Kirtland AFB, N.M. Distribution of this document is limited because of the technology discussed.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Air Force Weapons Laboratory (WLRT)
Kirtland Air Force Base, New Mexico 87117

13. ABSTRACT

A group of SPUTTER routines has been modified or developed to execute in a more nearly accurate and efficient manner the task of "rezoning" the Lagrangian zone structure in a specified region of the system being analyzed. The automatic routine AUTORZ has been expanded to provide for maintaining a prescribed spatial definition within a region. The control routines REZONE, CONVER, and ZONE have been modified to make use of the new routines ZMS and ZRAD, which accomplish conversion between space and mass coordinates, and a special procedure PROFL is described which performs energy interpolation in certain cases. To facilitate maintenance of complex computer codes such as SPUTTER, OIL, and HECTIC, two FORTRAN analysis routines (FAR, Programs One and Two) have been developed and are nearly complete at this time. These routines are designed for replacement of FORTRAN symbol strings by others, and for compilation of complete cross reference listings of variable and subroutine references. Editing routines designed for specific use with the data-link terminal facilities now installed at the General Atomic Special Nuclear Effects Laboratory (SNEL) and at AFWL are also described.

(Distribution Limitation Statement No. 2)

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Data link routines FORTRAN analysis routines						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.